

September 1988

\$4.50 U.S.

\$5.95 Canada

ST Applications

THE ATARI ST JOURNAL



LateST News • Software Reviews • Best and Worst

WordPerfect Printer Utility • Desktop Configuration Selector

The Hardware Column • Bunky's EZ-Print • Chaotic Equations



Now in our 4th Year of publication

ST Applications

THE ATARI ST JOURNAL

Editor and Publisher: William Petry

Assistant Editors: Sharon Richmond
Nancy Koepp

Photographer: Martin Majano

Subscriptions: Amy McLintock

Assistant to the Publisher: Michelle Stevens

ST Applications- *The Atari ST Journal* (ISSN 0887-3755), is copyright © 1988, and published monthly by **ST Applications**, 9890 River Road, P.O. Box 980, Forestville, CA 95436 USA. Phone (707) 887-7879. Editorial office at 9890 River Road, Forestville, CA 95436. Hours: 8:30 A.M.- 5:00 P.M., Pacific Time, Monday thru Friday.

Subscriptions: United States subscriptions are \$32.50 per year. Foreign subscriptions are \$40 per year (\$78 for Air Mail) and payable by Check on a US Bank or International Money Order *only*. For subscriber and advertising service please call (707) 887-7879.

Back Issues: Issues prior to June 1988 are available for \$3 each (\$8 with disk). Twelve issues of 1987 are available for \$30.00. All prices postpaid.

ST Applications is in no way affiliated with Atari Corporation.

Atari, 520ST, 1040ST and the Atari logo are registered trademarks of Atari Corp.

GEM is a registered trademark of Digital Research Inc.

Views expressed herein are those of the authors and may not necessarily reflect the views of **ST Applications** or its staff. To the best of our knowledge the information contained herein is correct. However, **ST Applications**, its editors and staff cannot be held responsible for the use of the information or any damages which may result.

Second-Class Postage Rates paid at Forestville, CA 95436.

Postmaster: Send address changes to **ST Applications**, P.O. Box 980, Forestville, CA 95436.

Contents...

2 EDITORIAL

3 PUBLISH YOUR PROGRAM

by David B. Smith
Get Published!

5 ADULTS HAVE MORE EXPENSIVE TOYS THAN CHILDREN

by Sol Guber
Laser Printers

7 WORDPERFECT'S PRINTER PROGRAM UTILITY

by Susan Robison
An Easy Approach

10 DESKTOP PUBLISHING WITHOUT THE PRICE

by Richard Caldwell
Migraph's DeskJet Driver

12 DESKTOP CONFIGURATION SELECTOR



by David T. Jarvis
Boot-up Flexibility

16 BUNKY'S EZ-PRINT



by Hugh McLean
Novices' Guide to Pascal

19 CHAOS WITH YOUR COMPUTER



by Sol Guber
Chaotic Equations

24 OVERLAPPING WINDOWS AND REDRAWS IN PASCAL



by Bruce Wiebe
Program Redraw

Miscellaneous File

53 ADVERTISER INDEX

2 THE HELP FILE

72 NEW SOFTWARE ETCETERA

Departments

29 THE GAMEMASTER

by R. Bradley Andrews
Slaygon and Gunship

31 LATEST NEWS

by Stephen Mortimer
Recently Released

33 NOT SO,BASIC

by Eric Thornton
Pickfile



35 THE HARDWARE COLUMN

By Bob McDonald
Questions and Answers

37 DESKTOP PUBLISHING

by Dan Fruchey
Timeworks' Publisher ST

42 THE BEST AND THE WORST

by Dick Biow
WordPerfect, HP DeskJet, Piracy
and more

47 THE JOYS OF MODULA-2

by Sol Guber
NeoChrome



50 PROLOGOMENON

by Joseph Schmuller
Parse Trees



55 STIPS.PAS

by Timothy Magee and Compulogic
Development Corp.
Simple File Copying Program



59 PENNSYLVANIA 6-8000

by Eric Thornton
Mumbling Mnemonics II



Indicates item(s) will be found
on this month's Support Disk.

The Review Board

60 Indiana Jones and the Temple of Doom- U.S.Gold Limited

62 Connect- ST Systems

63 Interlink-ST- Intersect Software Corp.

65 Secrets of Dungeon Mystery- Software Heaven/FTL Games

66 GFA BASIC Training and Reboot Camp- Michtron

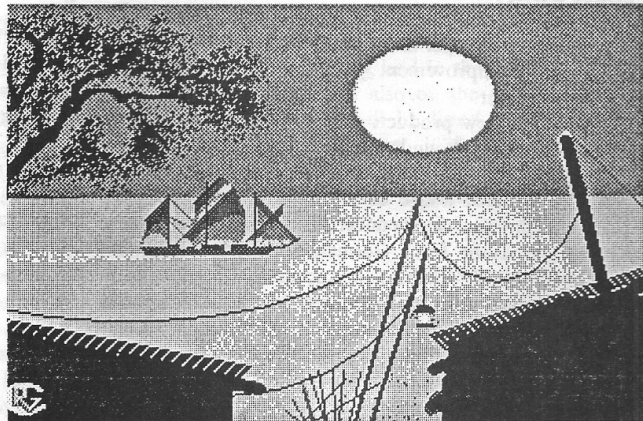
67 Santa Paravia and Fiumaccio- TDC Distributions

68 Tanglewood- Microdeal/Michtron

69 Switch/Back- Alpha Software

70 Rockford- Arcadia Software

71 Questron II- Strategic Simulations, Inc. (SSI)



This Month's Cover:

This month's cover is from public
domain Boat.Neo. The 2 color
separation was created by Eric
Thornton's Plater Program.

Editorial

With this issue we begin our fourth year of publication. I want to personally thank each and every one of our subscribers and readers for keeping us on track and still on the scene. We gladly accept comments and criticism and any changes you feel may improve our coverage of the Atari ST.

By now you may have noticed that there is no ST Business section this month. Although still bullish on the Atari ST, Don & Carole Terp are again resigned to the fact that the Atari marketplace as of yet cannot adequately support their endeavours.

Our agreement to include ST Business in ST Applications was an effort to both fulfill subscription obligations from the original 2 issues of ST Business (Sep-Oct '86, and Dec '86- Jan '87) and try to see if the Atari ST marketplace was as sound and progressive as we thought it may well be to get ST Business back on its feet and independent again. Should you enjoy the business tips and information feel free to inquire about the Val World Gazette for the Epson QX-10 computer that Don and Carole still publish. Their address is 5140 Appletree Drive, Roanoke, VA 24019.

The Atari marketplace (and other computers as well) has remained soft (ie. slow sales) since last winter and prospects for improvement seem rather uncertain for a boom this winter.

Promises of new products and a fall advertising push in the US still abound. My feelings are that until product is on the dealer's shelves and the buying audience is made aware of this fact the prospect of a bountiful Christmas looks uncertain.

A recent visitor from Germany brought cheery optimism of the ST's continued success in Europe. New professional software and hardware (released over there first) seems to be arriving regularly (of course all the documentation is in German or what have you).

Thanks for the past three years,

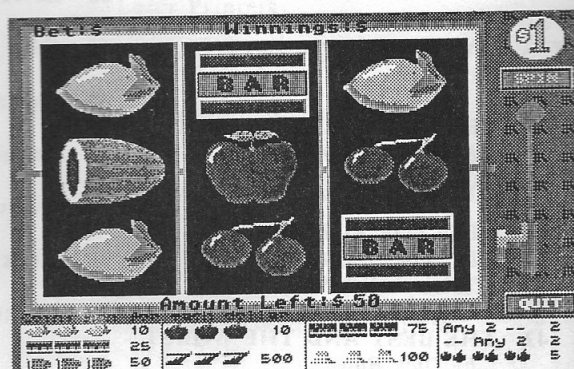
Bill Petry
Bill Petry
Publisher

TheHelpFile

Two problems were found in listings in the July 1988 issue.

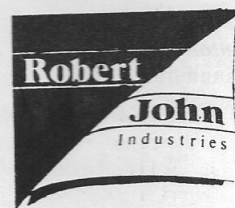
On page 50 the listing "Balldrop.bas" was missing the first line. It should read:

```
1 openw 2:fullw 2:clearw 2
```



In the listing for Super Slot Machine starting on page 22 besides the "Dat1" and "Dat2" files referred to in the left column not being included in the listings, four lines of program are missing per page of print throughout the listing. The 32k picture files for the slot machine display (see picture above). and a correct copy of the source code are available to anyone wishing a copy by sending an SASE (45¢ postage please) with disk and we will send them to you return mail. Sorry for any inconvenience this oversight may have caused.

Printed by:



5351 Oceanus Drive
Huntington Beach, CA 92649



Publish Your Program

By David B. Smith

Why not get in print!

Do you have an original utility, business application, game or other ST program? Perhaps you've made a successful modification of public domain material. Maybe you have an article in mind that would be of interest to ST users. If you have such a program, why not get it out, dust it off, write it up, and send it in to **ST Applications!** Here's how.

If you aren't too sure that writing is your long suit, or you think there isn't much chance of being accepted, you might be wrong! My late father thought a camping article he wrote didn't have much chance of being printed, but it was published in a national sporting magazine. For a year, Charles T. Smith was "in" every barber shop in the country. If you have something informative for ST users, your chances of being published in **ST Applications** might be a great deal better than you first may think!

Your first step is to decide that you want to share your program. Then, you need to take the plunge and write it up. This magazine is my choice since it is published monthly and continuously in quality magazine format.

And it is timely. If you do get in print, it will be soon, not six months or a year from now, as is usually the case. It's a fine magazine to show to your friends when your article is published!

There are many satisfactions in having your program and article published. The stipend isn't your main con-

sideration, but it will get you a subscription renewal or a year of back-issues. Your name will appear on the article, in your program listing, and in the Table of Contents of the magazine, and also on the monthly disk's directory! Your program will be listed in the annual directory, too. See the February, 1988, issue. And, just maybe, your program might make the bottom of the front cover! It happens once in a while with feature writers, even though the Departments get there more regularly!

If you are concerned about syntax or grammar, you surely have a friend that can buzz up your copy and rework the phrasing for you. Your readers will appreciate clear, simply worded style, and will pay attention to what you have to say more than fancy prose footwork. You needn't try for cutesy style or thundering literary phrases. Instead, be logical and start from the beginning, go step-by-step, and explain everything as simply and clearly as you know how. Every month's issue has readers who are just starting out with their STs. Then, rewrite, rewrite, rewrite. That's how you produce an article of professional quality. I might write a paper in and hour or two, and then spend a week of similar sessions rewriting it!

Your program, in whatever computer language you are using, should have remark (REM) statements throughout. I'm doing this more faithfully as I go along,

myself. What seems clear to you might not be clear to a new programmer.

If you would like ideas for style of writing, reread the last dozen issues of *ST Applications*. I program in GFA Basic so I follow programs by *Eric Thorton*, *Len Shikowitz*, and *Roger L. Yap*, just to name a few. Look at Roger's "Super Slot Machine" in the July, 1988 issue!

Your article should spell out clearly: 1) why you wrote the program, 2) interesting nuggets that happened to you on the way to the typewriter, 3) what the program does, and maybe does better, 4) a description of its operation right down the logic flow (yes, even though you have REM statements), 5) a repeat of important logic lines such as formulas. Ex.:

Smith formula for parallel Z; one arm (R2) resistive:

$$\text{Tan Delta} = X \cdot R2 / R1 \cdot R2 + X \cdot X + R1 \cdot R1$$

$$Z \text{ total} = (X + R1 \cdot R1 / X) \cdot \sin \text{Delta} \cdot (\cos \text{Delta} + j \cdot \sin \text{Delta})$$

6) interesting applications, anecdotes, caveates, hints, 7) and a short summary, and shorter sign-off. Change any of this to suit your style and add whatever you think would make interesting reading. What would you tell a friend in your home about your experiences when drafting your program? How would you give him/her directions in using it? What would you add as additional conversation?

By the way, don't worry if there are other similar, yet different, programs out on the bulletin boards. Readers of any magazine, ours included, have not tried everything there is. No one can. Some users don't have modems nor extensive PD (public domain) libraries. Most of us don't read every ST magazine and newspaper. But we read this one! Anyways, your special variation may be the one that readers have been waiting for!

Don't worry about criticism, either. Some of my program lines over the past year could have drawn fire from advanced programmers. But guess what! No one has ever complained! I know the pros out there spot my clutzy code, the same as I eventually do (I've only used GFA for a year). Our "special" readers know that we publish for them, and they are very positive. On the other hand, you may not get a bunch of fan mail either.

Print your article double-spaced using LQ or NLQ. I don't justify right margins. Tell *ST Applications* what word processor you use, and send a copy of the article on the disk you submit. Since Assistant Editor Sharon Richmond has never complained, I don't send articles in ASCII (by the way, let's put in a warm hello to Assistant Editor Nancy Koepp right here). Include a print-out of the program. Put the program on the disk with the source code, and of course a compiled version (if available) so everyone can run it. *ST Applications* is famous for remembering new-comers to ST computing, so your ST Basic efforts will be just as welcomed as other languages. Include a very short cover letter. No need for a sales pitch. Include any illustrations, such as a graph

print-out. Include a title page with your article. Follow the submission information found in each issue. And don't worry about security. The folks in Forestville are totally reliable.

You also will find that you have great freedom of content. If you want to include miscellaneous or divergent items, all you have to do is read any of my articles to see how lenient and forgiving the editors are.

For instance here is a discovery I'll include in this article. I found that I can perfectly transfer my Word Writer text files into ST Writer files! Maybe this is old hat, but if so, I've never known it. I change them to ASCII and copy them to the ST Writer file disk. When loaded, ST Writer automatically makes the conversion into standard format! The Smith formula used above is my own, is completely original and is published for the first time (to the best of my knowledge). I've checked engineering formulation texts. (It reduces 16 steps to just two steps in calculating the impedance of a parallel resonant circuit where one arm is resistive). One sour little electronic correspondence school I offered it to said the denominator was "too big"! They couldn't understand my derivation. See how I snuck it in here? This is one of the joys of publishing in "The Atari ST Journal."

ST Applications is very professional and seems to welcome scholarly thought. They won't needlessly redo your article. If you are unpublished, I assure you this is not the norm in the print media.

Talking about digressing, our continuing battles with mail order firms go on. A Massachusetts "ST" used software firm offered to trade original computer programs. You send the original program, docs, package and a fee. You then trade for something else similarly received. I sent my first order in February, and the second a month later. As of July 7, I have not received an answer to either order nor to two subsequent inquiries (one being a certified, receipt requested notice). An attempt to phone got a non-working number message. I sent eight of my original ST programs, along with over fifty dollars. I cannot recommend them.

Customer Service at MCI MAIL is A+; just outstanding! I hope you don't have to try the OAG customer service. I've just received some credit card line-of-credit checks from the good people at Apple Bank in Garden City, N.Y. You probably have this service, but the banks on Guam don't. Now I can order software and other computer items by mail using these checks. The great advantage is that I don't have to release my credit card expiration date to the computer dealer. This way, the dealer can't charge my credit card account without my knowledge nor permission as a California computer firm did (see my October, 1987, article on mailorder).

Continued on page ==> 11



Adults Have

\$ \$ \$

More Expensive Toys

\$ \$

Than Children

by Sol Guber

No this is not a review about games for the ATARI, it is an accounting of my experiences purchasing a laser printer. Ever since ATARI announced their laser printer for under \$2000., I have been coveting one. The object of any computer is to perform useful work, and even though floppy disks have been around for more than ten years, the human race has not had a chance to generate the evolutionary adaptations needed to read them without mechanical equipment. For this reason, along with computers, you need monitors and printers. Yes, you can generate a beautiful picture or drawing and send it to all of your friends along the modem's path or with a floppy disk, but you need hard copy to show to your mother, who wants to take it to HER friends to show what an intelligent child she has.

So now that I have gone into the epistemology to show that printers are almost a necessity, I need to describe, why a laser printer is almost a necessity. Many years ago, when the typewriter was invented, it was deemed a travesty to perform any sort of communications with it. In those days, a good pen was all that was needed, and only for crass businesses was a typewriter useful, because time is money and a typewriter was much faster than writing by hand. When the printer was invented, the daisy wheel version was thought to be the epitome of chicness since it looked much like a typewritten version. Once the dot matrix printer was produced in such

quantities that it was common, it was looked down upon, since it could not produce 'letter quality' type, but could produce some nifty looking graphics. With the coming of 24 pin matrix printers, even this argument was reduced since it could produce letter quality type and graphics. However, the 24 pins moving slowly to generate the letter quality also generated a very high pitched whine that drove many people to distraction.

Finally the laser printer entered the scene. It gave off a slight hum when it was not working and a soft burp as each piece of paper was ejected from its maw. The graphics quality was 300 dots per inch. It was also very fast. Even though the majority of the laser printers are rated at six pages per minute, it was never 10 seconds between pages; this number was for the second page. To generate the first page takes however long it is necessary to send the information to the printer, and for some applications, this is a significant amount of time. But the second and subsequent copies are really so fast that you wonder how it can be done.

I did not buy the ATARI laser printer for several reasons. The first was compatibility. The ATARI laser printer is a 'dumb' one that needs another system to run it. It contains no internal memory and expects all of the information to be sent to the printer from the computer. The information to the printer is sent through the hard disk port and is of blinding speed and can achieve 8

pages a minute from the very first page. It is run by a program inside the computer that needs to be loaded every time the computer is turned on. The printer knows how to be a Diablo 630 printer and how to work from GDOS drivers. Let me explain this in some more detail.

A portion of the TOS details how output is to be sent to various devices. When TOS was first written, it 'knew' how to write to floppy and hard disk, EPSON printers, modems, and MIDI devices. Since all the possibilities were not exhausted, there was another portion written called GDOS that could send the proper information to all other devices and one of them was the laser printer. There are several programs that use this to good advantage including EZDRAW and Microsoft WORD. However, there are many programs that do not use GDOS like Publishing Partner, and Certificate Maker. Since I had an off brand printer, I have had many problems getting printer drivers to be able to use both the program and the printer.

Which gets me back to my laser printer. I did decide to purchase the Panasonic Laser Partner. The major reason was that the Panasonic had five 'different' emulation modes, a Hewlett Packard Laser Jet, the EPSON FX286, the IBM ProPrinter, the Diablo 630, and the Panasonic KXP1092i. Each of these was available by pressing the button on the front panel and the printer now understood the various codes for each of these printers.

This gave me a great deal of flexibility. Now Graphic Artist could use the HP laser driver. Word Perfect now had a choice of Diablo or Laser Jet emulations. Certificate Maker could use the FX driver and many other programs would see a transparent printer. Did this work?

Well, when I got the printer across my threshold, it was in a box 2 foot by 2 foot by 2 foot. It was well packed and had many boxes inside of the large package. There was the toner, the developer, the drum, and on top of everything was the manual. I have looked at the HP laser jet manual and was very much impressed with it. I was disappointed with the Panasonic one, but only in comparison. It seemed about 2/3 of the size, but it was very complete. Reading the manual before I did anything, I then unpacked the various pieces. The various pieces fit together very easily (once I pulled off the proper pieces of plastic) and the finger screws locked everything well. The ink came in a plastic bottle that needed to be put into the top hopper. I thought that this was a cheap and messy method, but I did not make a mess, and I discovered that this bottle was half the price of a comparative amount of HP cartridge material. It took less than 10 minutes to put everything together. I plugged in the unit, and the front panel lit up saying that it was warming up. Once the ready appeared, I took the system offline and hit the print test button. I waited, and waited, and waited until a loud voice told me "Press the enter button". Three seconds later, out

came the first sheet. It was so much fun that I pressed the print test button (and ENTER) and another sheet came out. After this was repeated five times, my wife asked what else could it do?

Now it was time to hook it up to the computer. I unplugged the old printer, and wished it a good home, wherever it was sold to, and put the plug into the centronics plug on the back. I then went to the front panel and told the printer that it was a parallel printer, not a serial one. Turned on the computer, turned on the printer, got a small text file in one window and told the computer to print it. The light on the laser printer started to flash on and off, the screen stated that it was finished, and NO paper came out. My wife asked what else could it do? Reading further in the manual I discovered that there was data in the machine, but no form feed was given, so no paper could come out. I put the system off line, and pressed form feed, and out came my first computer generated laser printed piece of paper, that my mother could take to show all of her friends.

Now my daughter said, could I use it with Print Master to make up some invitations for a Bird Day party for her conure parrot. NO, I yelled, I need to convert all of the printer drivers. When she burst into tears, and my wife asked what else the printer could do, I told her to wait and I would put the right printer driver onto the disk and she could use the Print Master in 5 minutes. It did not take that long. I pushed the buttons for the proper emulation, copied the printer driver onto the disk, and away she went. The printer worked beautifully for the first copy. We set the number of copies to 10, told Print Master to send out the invitations again, and 90 seconds later, had all the copies that we needed. Everybody was duly impressed.

After the invitations were in the mail, I looked closely at what I had printed out. There were two flaws that were not so obvious at first appearance. The first was that the graphics were a bit chunky. The dots were not nice rounded dots from the pins, but were square boxes that looked a little bit worse then I had expected. I went and pulled out some other work that I had done using Print Master and compared it to the laser printer. My memory was worse then I thought. The printing was worse on a 'real printer' but my expectations were not quite so high. The second problem was that the border on the card had about 1/8 inch cut off. This problem was much more serious and when I printed out the card again, the same thing happened. Back to the printer manual to see what the problem might be, to see if I could feed the paper forward slightly.

There was no way to move the paper slightly to the 'print head' and the problem was more serious then I thought. A piece of paper is 8.5 by 11.0 inches. The laser drum system does not print on the whole piece of paper.

Continued on page ==> 9



An Easy Approach To WordPerfect's Printer Program Utility

(Editor's note: Susan Robison is an Atari Printer Specialist at WordPerfect Corporation. Her job is to write printer drivers for the Atari ST. She does this using a program called PRINTER.PRГ, which is the same program that is distributed free of charge with every copy of WordPerfect. In theory, anyone using PRINTER.PRГ can teach WordPerfect to drive any printer ever made, or he can customize any WordPerfect driver to suit his own taste regarding such features as letter-spacing, darkness of emphasized type, and underline style. In real life, most folks are too awed by the program and too intimidated by its rather curt manual to dare assault their printer drivers with it.

We asked Susan to write a "follow me" sort of article that would walk our readers through the operation of PRINTER.PRГ far enough to establish confidence. If you have been wanting to use PRINTER.PRГ but are uncertain about its operation, we encourage you to sit down at your computer and follow Susan's lead as you read the text. If not -- and even if you don't use WordPerfect -- we urge you to file this article carefully. One day, you are going to need it!)

by Susan Robison

WordPerfect's Printer Program is the interface between your printer and WP. Since every printer uses codes unique to itself to communicate with software, the Printer Program allows you to create, modify or review a printer 'driver' that is specific to your printer. If you make any modifications to the printer driver supplied by WP, I would suggest that you write down any changes you make as you go. Then if your changes don't work, you can put the original codes back in. Also, WP's customer support department may not be able to help you with problems you may have with your own driver or one you have modified. If you call for help and this is the case, be sure and tell them of any changes you may have made. Please be aware also that some features are left out of your Printer Driver because of printer limitations.

Printer Program can seem a little forbidding the first time you venture into it and look around. The program itself is actually simple to use but the strange combinations of codes are confusing. The codes for your printer are found in your printer's Technical Reference Manual. The manual may list the codes for each function in as many as four different ways: 1) as a Control Code or Escape Sequence, 2) in HEX format, 3) in

Decimal format or, 4) in BASIC syntax. (See Printer.Man for explanation of BASIC, beginning Pg. 1, line 38.) WordPerfect uses Control Codes or Escape Sequences. However, the abbreviations in Escape Sequences must be translated into Decimal values, i.e. ESC = 27, BS (back space) = 8, LF (line feed) = 10. The Decimal values will be in your manual and should be entered into Printer Program with corner brackets: <27>, <8>, <10>.

Supplied with every WP Learn diskette is a file called Printer.Man. It would be helpful for you to print this file out and bind it as a reference manual. It starts with a Table of Contents, ends with an Index and has 27 pages of information and explanations of the program and Special Function Codes. Because of changes to the Printer.Man file on the 8-1-88 release of WP, the page numbers referred to here will be incorrect for 4-15-88 and earlier releases although the information needed is still there. I will lay out step-by-step instructions for making changes in Printer Program but will refer you to this file for more detailed explanations of codes and the program itself.

For my examples, I will use the driver for the HP LaserJet B cartridge. If you step through your Printer Program as you read, it will be easier to follow.

Modifying The Driver

If you wanted to change the way your printer marked Redlined text, you would follow these steps:

1. From DeskTop, double click on Printer.Prg. You will come to the initial Printer Program Menu.
2. Select option 3 by clicking on it or pressing a 3 on your keyboard. The list of available printer definitions will come up with several options listed. (See Printer.Man pages 6-8 for explanation of options.)
3. There are several ways of finding the definition for your printer in the list:
 - a. Drag the slide bar. It works the same as it does in WP.
 - b. Type in the number of your definition and press enter. You would have had to taken notice of the number while selecting your printer in WP.
 - c. Press (S)earch (one of the options to the side) and begin typing in the name of the definition.

Select the "LaserJt Reg, +,500+ B: Tms Rmn1". If you are using a 4-15-88 or earlier release of WP Print and Font disks, the definition is "LaserJt +,500+ B: Tms Rmn1". When you find the correct definition, highlight it and click on it or select (E)dit from the options listed. The "Edit Printer Definitions" menu will come up. There are 11 screens or catagories in this menu; 1-9,A and B.

4. Select option 7, "Special Text Markings". On this screen, there are three lines pertaining to the Redline function. #6 is "Redline Character Code", #7 is "Redline On" and #8 is "Redline Off". WP will automatically do Redlining if you have a Character Code selected. Numbers 7 and 8 would be left blank. In the case of the LaserJet B definition, Redline is done by Italicizing text. The Character Code is set to 0 (no character specified). Number 7 has the codes <O> <2> in it. The special function <O> tells WP to save the current font number and switch to a new font. The <2> is a binary code equal to the number of the font desired, in other words, font 2. (See Printer.Man pg. 4, ln. 26)

Number 8 has the code <P>. This tells WP to switch back to the original font selected when Redlining is finished. (See Printer.Man pg.4 ln. 37.)

We will change the Redlined text to be marked by a character along the left margin rather than by Italics. Delete the codes in numbers 7 and 8 and enter the Character Code for the character you want to use in number 6. For example: in the character table for font 1, a vertical line is character 124 and a plus symbol is number 43. These are the most commonly used Redline characters. Select #6, type 124 and press enter. I will explain how to view your character tables later.

5. You can then exit the "Special Text Markings" screen by pressing F7 or clicking on the EXIT box. You will be returned to the "Edit Printer Definition" menu where you can select another screen to make other changes or you can exit again. If you opt to exit the Edit menu for this driver, you will first be given an option to save your changes. A message will appear

that says "This Definition has been Modified. Do you want to save it?" The name of the definition will be listed below and three boxes with "Yes", "No" and "Cancel". Select "Yes" to save the changes you have made. The driver will be updated and you will be exited to the full list of printer Definitions. Exit again to return to the main menu.

Modifying A Character Table

You may want to make changes to characters in a character table. There are two ways to access the character tables:

1. If you don't know the name of the Table, you can access it through the printer driver by following steps 1-3 in "Modifying the Driver" and then selecting B "Select Character Tables". This screen displays the Tables for all 8 fonts.

2. Highlight the table you want to access by cursoring to it or clicking on it. We want to access the table for Font 1 and it is already highlighted so select option 9 "Examine or Edit". You will see a list of all available fonts with a slide bar and a list of options. (See Printer.Man pgs. 22-25 for details on Character Tables and Options). The table you selected should be highlighted on the menu.

3. Select the Character Table by clicking on it or pressing E for [E]dit on the keyboard.

There is a much shorter way to access the table if you already know the name of it. Exit back out to the main menu:

1. Select option 4 "Character Tables" from the main menu. The entire list of available tables comes up.

2. Highlight the table called LsrJt-B Normal and select it. You will enter the Character table.

There are two kinds of Character Tables: Microspace (MS) and HMI w/Adjust. (See Printer.Man pages 14 and 22). LsrJt-B Normal is a MS table. There are three columns of information for each character: Display, Prt and Width. Display shows the character and its Decimal value. Prt is the instruction telling the printer how to print it and Width is how many units of space the character needs to print within. On the bottom of the screen it says "Note: Character widths are shown in 1/300ths of an inch". Some tables will be measured in 1/120ths or another unit of measure. This corresponds with question 4-1 (screen 4, question 1) in the Printer Definition.

You should also note that character widths are only used by WP if the table is in a proportionally spaced (PS) font. A non-PS font uses Fixed Pitch which gives every character the same amount of room regardless of it's size. LsrJt-B Normal is in a PS font. WordPerfect reads the widths entered here to calculate it's movement for underline, bold, overstrike, strikeout, etc.

If the character table is HMI w/adjust, you can control the room each character is given yourself. The table will show the same information the MS table does with the addition of a column labeled "ad". This is for an ad-

just factor. The width you enter will determine the exact amount of room the character will be given to print within and the adjust factor will move the character within the allowed space.

Exit the character table and select the IBM Graphics table from the list.

For an example, I will suppose that the space between words in my printed text is too small. I want to widen the Space character.

1. Cursor to or click on character 32. This is the Space character.

2. With character 32 highlighted, press B or click on option B "Character Width" below. You can widen the Space character by increasing the number. I would suggest adding 2 to 4 units. You may need to print some text from WP to see if it looks how you want it to. If you had been editing a printed character, you could also move it to the right of it's allowed space by increasing the adjust factor and to the left by decreasing it. Adjust factor can be set no higher than +7 and no lower than -8.

3. When you are finished making the adjustment you want, exit out of Printer Program and answer "Yes" to save your changes when asked.

When you become familiar with your printer and it's codes, you can 'customize' your driver or create drivers of your own. The Special Function codes listed in Printer.Man are familiar to all printers with the exception of the Apple LaserWriter and other printers that use PostScript rather than XonXoff protocol or Hardware Handshaking. The control codes vary from printer to printer. For example: A <28> may be used as part of an Italics code for one printer and a Subscript code for another. In order to write or modify your printer driver, you will need to become familiar with the manual that came with your printer.

WordPerfect Corp., unlike most Software companies, has included access to the printer drivers to allow users who have a need or interest in customizing drivers to be able to do so. Use the Printer Program carefully and good luck!

END

More Expensive Toys from page 6

It leaves a border of about 1/5 of an inch on all four sides. This must be taken into account when the program prints and the ability to tell the printer where the edge of the paper really is, is one of the options that is available with the Panasonic Laser Partner. I made slight modifications to the border and this time the printer worked much better.

The next program that I tried to get to work was *Certificate Maker*. Having had all of the experiences with *Print Master*, *Certificate Maker* was a breeze. I chose the Epson FX print driver and it worked on the first try. No problems with the border of the pages and the quality seemed to be higher than with *Print Master*.

The next program that I tried was *Graphic Artist*. One reason that I did not buy the ATARI Laser printer was that it would be very difficult to write my own printer driver for *Graphic Artist*. I feel that *Graphic Artist* is a very underrated CAD program. I have used it for several years and have used it both for CAD applications and for small amounts of business graphics and have become very accustomed to the program. It has a cumbersome interface, but generates nice effects. I used the HP draft mode printer driver and had a perfect copy the first time. A comparison with my old printer showed that the effort was easily worth the money.

The final test was *WordPerfect*. I changed the emulation to HP Jet on the front of the machine and then got an old review out. I changed the font to 5, which should print things horizontally in the IBM version. It did not work! This was my first disappointment with the Printer. Being technically minded, and being able to borrow a Laser Jet printer manual, I tried to write a printer driver for *WordPerfect* using their printer driver routine.

I looked at what *WordPerfect* did, what Panasonic did, and the codes for the Laser Jet. It seems that about 85% of the codes are the same. The HP laser jet has several built in fonts that the Panasonic does not have.

It seems that during the initialization of the printer, several codes were sent to select fonts. Once this code was deleted, I tried it again. This time there was no problem printing horizontal. The only thing that I was missing was the BOLD font. Are there any problems with the Panasonic? Yes, it does hum continuously and gives a little sigh every 30 seconds or so as the toner settles itself. The manual is above average, but there are many, many things that are not fully explained. The areas that need much more explanation are the creation and downloading of macros. You can also tell the printer to put the gray scale and hatch pattern that can be set for certain areas of the page and then print over these areas. There is also the ability to print graphics patterns by using vector graphics to generate the patterns inside the printers memory in any of three different dots resolution. Finally, the ability to download fonts is also available, but it will take a great deal of programming ability to change the fonts into your own. (In comparison, the HP manual states that the technical manual explains all of these features and is available for \$30.) I am extremely pleased with the Panasonic Laser Partner. It is fast and clean. With the various emulations, every piece of software that I have tried was able to work once the proper printer driver was found. It is easy to change the emulations and to set up the printer the way it is needed. I will give you further reports on my progress with the printer.

END



Desktop Publishing Without the Price

By Richard Caldwell

Old Slogans Never Die...

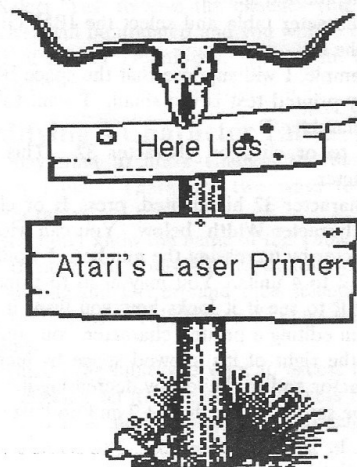
As slogans go, "Power Without the Price" has gotten pretty threadbare. Who can say, "Power Without the Price" these days without at least a little smirk ("Promises Without the Product" is just one of the many thoughts that bring about this nervous twitch.) But just when it looks like we should put our favorite phrase in moth balls, someone comes along gives it a new lease on life.

What we're talking about here is laser quality printing at half the price. You may have heard that promise from Atari before but this is another story. The Hewlett-Packard DeskJet is available for use with your Atari ST right now. You read that right. The HP DeskJet is an plain paper ink-jet printer that provides 300 dot per inch resolution.

You won't need 4 megabytes of memory or a hard disk to use the DeskJet and it is priced at less than half the price of Atari's laser printer. HP may not be the first company to come to mind when you hear, "Power without the price," but they seem to be offering just that with the DeskJet.

Dick Biow eloquently described the capabilities of the DeskJet as a text printer in *Best & Worst*. But what about graphics printing and desktop publishing? To get the most out of the DeskJet in graphics applications you will need some software to drive the DeskJet to its maximum resolution.

One of the first companies to provide such software is Migraph. Migraph was quick to spot the DeskJet's potential as a high resolution graphics printer. Migraph's



DeskJet driver is intended primarily for use with their Easy Draw drawing and page layout program. Since Easy Draw uses GDOS fonts and drivers the DeskJet driver should work with any program that uses GDOS properly. You'll need just a megabyte of memory and a pair of double sided drives to use the DeskJet driver.

All This and More...

The DeskJet driver includes GDOS fonts and drivers for both 300dpi printing and 150dpi draft mode printing. The fonts provided are the same Swiss (Helvetica), Dutch (Times Roman) and Typewriter (Courier) fonts and sizes that are provided with Easy Draw but are scaled to take full advantage of the DeskJet's resolution. Also provided is a screen dump utility and a program to set the ST's serial port to 19.2k bits per second. The latter is to speed printing for those who wish to use the DeskJet through their serial port. It's a nice touch for Migraph to allow for such a strange whim.

Once you've taken stock of these goodies you will have the unique pleasure of installing the fonts and drivers to work with your GDOS application. Cynics might say that this is very much like the unique pleasure of repeatedly banging on your forehead with a rubber mallet. I'll simply say that it can be a tedious process.

All of the font and driver files must be in the proper directory for access by GDOS and a file called ASSIGN.SYS must correctly reflect what fonts and drivers are available and where they reside on disk. Since this means moving a lot of large files around, a hard drive eases the task immensely.

If you follow the instructions carefully and take the installation step by step, you'll find that GDOS installation need not be painful. If you should begin to feel that rubber mallet sensation, Migraph provides excellent phone support and can probably help you extract yourself from virtually any mess you've gotten into.

Once the DeskJet fonts and drivers are installed you can just forget them and use Easy Draw or another compatible GDOS application (such as MS Write) as you would normally. Just remember that you now have a full 300 dots per inch of resolution to work with. This will free you to use font sizes and styles that might have been illegible on a dot matrix printer. You can also draw finer objects and use 300dpi scanned images without fear of them becoming strange black blobs.

When your page is complete and ready for printing, save the file and call up Migraph's OUTPRINT.PRGM program. Outprint is Migraph's enhanced version of Atari and Digital Research's Output program. Like Output, Outprint allows you to select one or more pages to be printed. In addition, Outprint allows you to change output drivers from within the program. This allows you to switch between the 150dpi draft mode and 300dpi DeskJet drivers.

Get a Cup of Coffee...


It won't take long to find out why this flexibility is so essential. In draft mode the Deskjet can print a full page of text and graphics in 4 or 5 minutes. This will give you a good idea of what your final result will look like when printed at 300dpi and might be acceptable for some applications. Be sure to look it over carefully, because the final draft will take 10 to 15 minutes to print. The DeskJet is clearly no threat to laser printers when it comes to print speed.

The DeskJet's speed should not be a problem in the low-volume applications for which it is intended. Simply start the DeskJet printing and find something else to do for a while. The DeskJet feeds cut sheets without the slightest tendency to jam, misalign or otherwise ruin your printing, so you don't have to watch it every instant

of its operation. If for some reason you want to watch the DeskJet print every line of every page, you can do so in complete comfort. You will not be assaulted by the constant scream of tiny pins hammering ink onto a hapless sheet of paper. The DeskJet's ink-jet engine is virtually silent, so you will hardly even notice its operation.

What Do You Think?

But the forte of the DeskJet is print quality. How good is it? Maybe you can tell me - this entire article was printed using Easy Draw and an HP DeskJet. I think you'll agree that most people would not notice the difference. Certainly if you compare a laser printed page to a DeskJet printed page side by side, you may see some differences in quality. The DeskJet sometimes shows fine lines in its grey tones which belie its line printed nature. On the other hand, the DeskJet often produces darker, more even blacks than most laser printers.

HP's DeskJet lists for \$995 (but is available for considerably less than that) and Migraph's DeskJet driver lists for \$49.95. Migraph is currently offering both the DeskJet and their driver as a package for \$949. Does this mean that HP and Migraph have finally brought "Power Without the Price" to the ST desktop publishing world? Well, at the very least they're offering "Power at a Considerably Better Price Than Anything Else So Far" In the real world - as opposed to the world of advertising jargon - you can't do much better than that! 

Migraph Inc.

720 S. 333rd Street (201)
Federal Way, WA. 98003
1 800 223-3729
206-838-4677

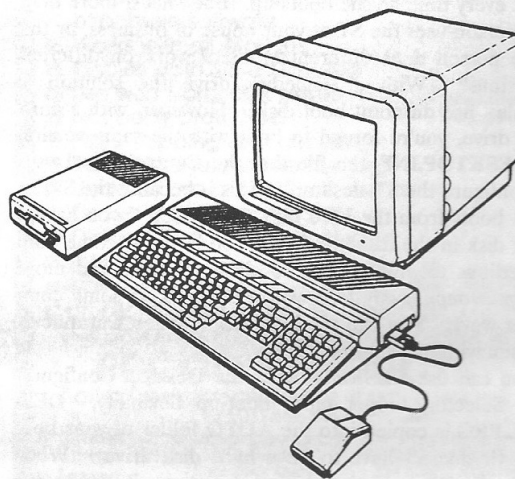
More Publish Your Program from page 4

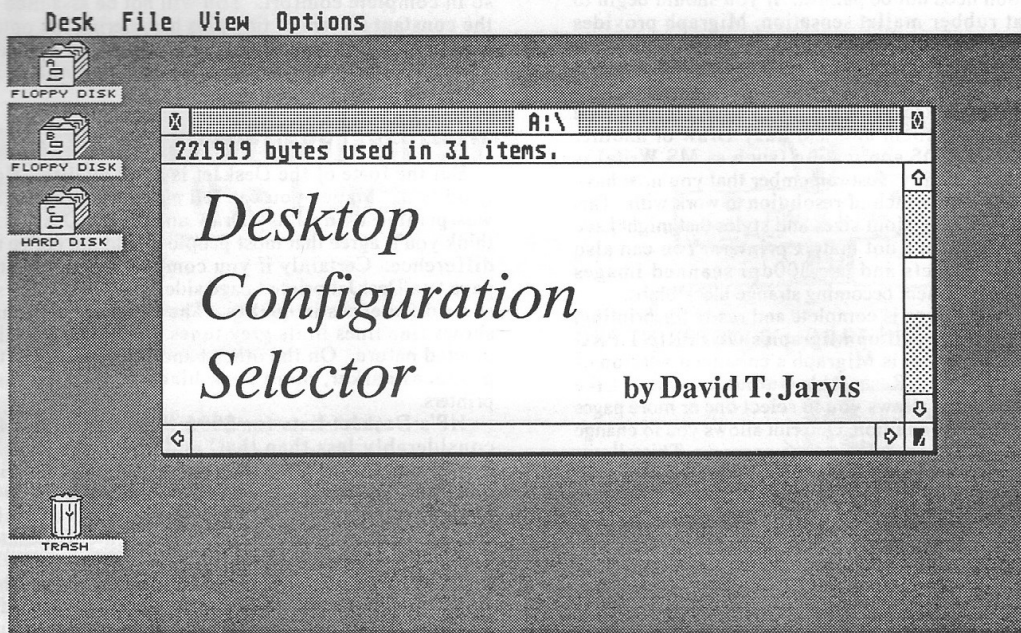
Computer Games Plus ships promptly and in the exact manner specified. We've had good luck with **Computer Creations**, and **Compute Ability**, also. Dealers, why aren't you placing your ads in this magazine where we can read them? I've been absent on these pages for a few months because I have been finishing my two year project, **Mystery Mansion ST**, hopefully soon to be released. Elephants take nearly two years to be born. I can appreciate their anxiety.

Well, there you are! If this article encourages you to get going and submit your super program, I'll be looking for it. So will thousands of others!

Publish that program! Cheers!

END





A DESKTOP FOR EVERY OCCASSION...

The Control Panel desk accessory for the ST gives you a great deal of flexibility and power in configuring the ST "Desktop": the resolution, colors, windows, and a number of other workspace attributes can all be set to suit you. By clicking on "Save Desktop", you guarantee that your ST will boot the same way next time. This works great for an individual who does the same type of work every time he/she boots up. But what if more than one person uses the ST at your house or business, or the same person does different types of work on different occasions? With a singledisk drive the solution is simple: use different boot disks. However, with a hard disk drive, you're forced to boot with the same version of **DESKTOP.INF**, the file that the Control Panel uses to configure the ST desktop. (This is because the ST always boots from the hard disk drive, even if you have a boot disk in the diskette drive.) So it can become kind of tedious to always have to close windows and move things around every time you wanted to do some computer work. You can see how this situation can quickly worsen with more users.

You can use **DESKSEL.PRg**, the Desktop Configuration Selector, to get more boot-up flexibility. **DESKSEL.PRg** is copied into the AUTO folder of your boot disk (it doesn't have to be a hard disk drive). When TOS, the ST operating system, boots up, it checks the

AUTO folder, if it exists; if any programs are found there, they are consecutively loaded and executed. Some have criticized the fact that this occurs before GEM (TOS's graphic interface) is loaded; this prevents you from setting up GEM programs to run automatically when you start the computer. In the case of the Desktop Configuration Selector, this proves to be an advantage. When **DESKSEL.PRg** runs, it checks the root folder, or directory of the boot disk for files beginning with the name **DESKTOP**, except **DESKTOP.INF**. Up to 10 of these are displayed, with the name of one of the ST ten function keys beside each. If the user presses one of the function keys displayed, the associated file is checked to make sure it has the format of a **DESKTOP.INF** type file; if so, then it is copied over **DESKTOP.INF** in the root directory. The program ends and GEM is loaded, with the desktop set up as indicated by the file you selected. **DESKSEL.PRg** also allows you to specify what folder the files are in; they don't have to be in the root directory. They can even be hidden files (hidden files don't show up in directory windows)! By having multiple "configuration files", each with a different file extension (for example, **DESKTOP.PRO** and **DESKTOP.WRI**) you can boot up with the appropriate "desktop" for the type of computer activity planned.

A DESKTOP.INF BY ANY OTHER NAME...

It's no wonder they call them "home" computers; they have windows, doors (for disks), trash cans, tables (some programs use them, anyway), and now you can have multiple desks, or "desktops". To set up the Desktop Configuration Selector on your boot disk drive, just follow these steps:

First, you need to create the different configuration files for each person or type of activity needed for your ST. Do this by setting up the ST the way you want it to look when you boot up, then select "Save Desktop" from the Options Menu. When you do this, TOS/GEM writes the current desktop attributes to a file called **DESKTOP.INF** in the root directory, in a format understood by the Control Panel program. Now, find that file in the root folder and copy it to another name, also beginning with **DESKTOP**, but with a different extension (the final (up to 3) characters, after the dot). Pick something that will remind you of the purpose, like your initials, or part of your name, or a description of the type of activity planned (for example, **DESKTOP.GMS** for the adolescents at your house who use the ST for games). As mentioned before, you don't have to put these files in the root folder, and you don't have to give them names starting with **DESKTOP**, but these are the default values that **DESKSEL.PRG** looks for first. If you want to "hide" your configuration files, put them in another folder and use a program such as "File Hider" (from the December 1986 issue of *COMPUTE!*'s Atari ST Disk & Magazine) to make them hidden files. Second, if you don't have an **AUTO** folder on your boot disk, create one with the New Folder option on the Desktop menu. Now copy **DESKSEL.PRG** into this folder.

That's it. Now reboot.

When **DESKSEL.PRG** runs, you have several options. If you don't want to change the current **DESKTOP.INF** file, press RETURN (or ESC). If you want to copy one of the files shown to **DESKTOP.INF**, press the function key shown next to it. If you know you have configuration files in another folder (or another disk), press **HELP** and enter a new "search string", consisting of an optional disk letter and colon, folder names (always separated by the backslash "\") and file specification, which may include the wildcard characters "*" (for one or more unknown characters) and/or "?" (for one unknown character). Examples of valid search strings include:

```
C:\DESKS\DESKTOP.*
A:\DESK???.*
B:\DAVID\DESKS\WORK\SECRET.*
D:\PUBLICDO\GAMES\ADVENT\WEIRD.INF
```

Note that the last example will only find one file.

AN ADDITIONAL FEATURE

If something happens to the **DESKTOP.INF** file in the root folder of your hard disk drive, GEM does -- or,

rather, doesn't do -- something funny. It won't boot. Apparently the Control Panel program crashes if it tries to read **DESKTOP.INF** and doesn't find it, or finds it to be invalid. You can use **DESKSEL.PRG** to fix this if it happens. Just use it to copy a valid **DESKTOP.INF** from one of your old diskette boot disks.

DESKTOP EX MACHINA (how it works)

DESKSEL.PRG is a good example of the fact that you don't necessarily need GEM to write an ST program. Written in C, it contains functions to control cursor location and character attributes on screen, and shows how to input "odd" character values, like function keys and the **HELP** key. **List_files()** is the function that does the directory searches (based on the contents of **search[]**, a character array) and places up to 10 file names on screen. If the user updates the search string, I just call **list_files()** again. **Locate()**, **clear_line()**, **rev_on()** and **rev_off()** are examples of how to use VT52 emulation supported by the ST. VT52 refers to a certain type of widely used computer terminal. Screen operations such as locating the cursor and clearing lines are activated by printing particular strings of characters, all beginning with the **ESCAPE** character (ASCII 27, in decimal), followed by a character indicating the type of operation, and, in certain cases, by one or more parameters. For more operation, see the book "Atari ST Internals", published by Abacus Software; it contains a description of these and several more operations possible with VT52 emulation.



MORE 

ProCopy ST BACKUP UTILITY

You can't backup your software because copy protection locks you out. **ProCopy** is the key!

- Protects against the accidental loss of expensive software
- Works with all Atari STs
- Copies both S/S & D/S disks
- Use with 1 or 2 disk drives
- Not copy protected
- **FREE** shipping and handling
- **TOLL-FREE** telephone number
- Updates are only \$12 with return
- Orders shipped same day
- Disk analyze function included

Dealer
Inquiries
Welcome

  and C.O.D. orders
\$ 34.95
Call (800) 843-1223

Send check for \$34.95 (overseas add \$2.00 for air mail) to:

PROCO PRODUCTS

P.O. BOX 665, CHEPACHET, RHODE ISLAND 02814 USA
(401) 568-8459

Available
Internationally
in Europe
and
Australasia

THE MOVING FINGER CO.
Building 2
Shamrock Quay
Southampton, SO1-1QL
England
Tel. 0703-227274

TECH-SOFT
COMPUTER WHOLESALE
324 Stirling Highway
Claremont
Western Australia 6010
Tel. 09-385-1885

ST Applications 13

```

/*----- Main Program -----*/
main()
{
    /* Initialize stuff */
    if (getrez())
        width = 80;
    else
        width = 40;
    search[0] = 'A' + Dgetrv();
    search[1] = ':';
    search[2] = '\0';
    strcat(search, "DESKTOP.*");
    /* directory search string */
    /* set search attributes */
    /* set Disk Transfer Address */
    /* Set up the screen */
    rev_on();
    txt_center(1, "Desktop Configuration File Selector");
    txt_right(2, "David Jarvis, 1988");
    /* function key indicated next */
    locate(4,0); printf("Press the function key indicated next\n");
    locate(4,0); printf("F01: Hidden files\n");
    locate(4,0); printf("F02: Hidden files\n");
    locate(4,0); printf("F03: Hidden files\n");
    locate(4,0); printf("F04: Hidden files\n");
    locate(4,0); printf("F05: Hidden files\n");
    locate(4,0); printf("F06: Hidden files\n");
    locate(4,0); printf("F07: Hidden files\n");
    locate(4,0); printf("F08: Hidden files\n");
    locate(4,0); printf("F09: Hidden files\n");
    locate(4,0); printf("F10: Hidden files\n");
    /* list up to MAXFILES files matching the search string */
    list_files();
    /* Get a character from the user and respond accordingly */
    done = 0;
    index = 0;
    do
    {
        longch = Bconln(2); /* use the LONG value to get scan codes */
        switch( (Bconln(1)) ) /* use the BYTE value to get chars */
        {
            case HELP:
                locate(23,0);
                clear_line();
                printf("New search string: ");
                gets(buffer);
                strcpy(search, buffer);
                list_files();
                break;

            case F01:
            case F02:
            case F03:
            case F04:
            case F05:
            case F06:
            case F07:
            case F08:
            case F09:
            case F10:
                index = (WORD)(longch-FUNCSTART) >> 16;
                if ((index > 2) || (index < 1))
                    cconout( BELL );
                else
                {
                    done = 1;
                    break;
                }
            default:
                if ((bytechar==CR) || (bytechar==ESC))
                    index = 0;
                else
                    done = 1;
                break;
        }
    } while (!done);
    /* if no file was specified, exit at once */
    if (!index)
        exit(0);
    /* If a file was specified, copy it to DESKTOP.INF */
    index--;
}

```

```

/*----- Desktop Configuration Selector -----*/
/* DESKSEL.C
/* David Jarvis, 1988
/* Version 021488
/*-----*/
#include <stdio.h>
#include <stdlib.h>
#include <portab.h>
/* Standard IO declarations
/* TOS bindings
/* to make code as portable as possible
/*-----*/
/* bit masks for directory attributes
/* read-only
/* hidden file
/* system file
/* volume label
/* directory (folder on the ST)
/* archival purposes
/*-----*/
/* Keyboard definitions (scan codes)
#define HELP 0x00200000
#define F01 0x003a0000
#define F02 0x003b0000
#define F03 0x003c0000
#define F04 0x003d0000
#define F05 0x003e0000
#define F06 0x003f0000
#define F07 0x00400000
#define F08 0x00410000
#define F09 0x00420000
#define F10 0x00430000
/*-----*/
/* Miscellaneous Definitions
#define MAXFILES 13
#define CR 13
#define ESC 27
#define BELL 7
#define STARTLIN 8
/*-----*/
/* Data structures & variables
struct data_str
{
    BYTE res[ 21 ];
    BYTE attr;
    BYTE time;
    WORD size;
    LONG size;
    BYTE name[ 13 ];
};
/* reserved by TOS
/* attribute byte
/* time stamped on file
/* date stamped on file
/* size in bytes of file
/* filename, extension & terminating 0
/*-----*/
/* search filespec (default: DESKTOP.*)
/* after current dir (A: \)
/* concatenate string to path
/* copies one string to another, max.len
/* copies one string to another, max.len
/* saves filespec shown on screen
/* configuration file character
/* configuration file line
/* string input buffer
/* used with Bconln() for directory listing
/* keep on going until this is 1
/* index to filename array
/* 80 or 40 -- width of screen
/* line count in file
/* number of lines in directory listing
/* return value of BIOS functions
/* Input character -- use LONG for scan code
/* old configuration file and new file
/* DTA used by this program
/* print file name at appropriate location
/* print file name at appropriate location
/* void locate(), clear_line(), rev_on(), rev_off(); /* cursor stuff
/* void list_files();
/* to get full pathname of input file
/* handles severe errors

```

```

/* Read the file through once (up to 100 lines) to
ensure that it seems to be a DESKTOP.INF type file */
if ( (cf = fopen(fnames[ index ], "r")) == NULL )
    sev_err( "cannot open file -- press any key" );
l_count = 0;
while ( (fgets(cf_line, 80, cf)) != NULL )
    l_count++;
if ( (cf_line[ 0 ] != '#') || (l_count > 100) )
    sev_err( "Incorrect file type - press any key" );
fclose( cf );

/* Open DESKTOP.INF for output */
if ( (inf = fopen("DESKTOP.INF", "w")) == NULL )
    sev_err( "cannot open DESKTOP.INF - press any key" );

/* Re-Open selected file for input */
if ( (cf = fopen(fnames[ index ], "r")) == NULL )
    sev_err( "cannot open file -- press any key" );

/* Read in the old file, one character at a time, and write to the new */
while ( (c = fgetc(cf)) != EOF )
    fputc(c, nf);
fclose( cf );
fclose( nf );

/* ----- Miscellaneous Functions ----- */

void list_files() /* display files matching spec */
{
    WORD i;

/* find FIRST configuration file on directory */
    flags = Ysfirst( search, s_attr );
    f_count = 0;

/* as long as no errors are returned, find NEXT configuration file */
    while ( (iflags) && (f_count < MAXFILES) )
    {
        if ( strcmp(dta.name, "DESKTOP.INF") )
            build_fname( search, fnames[ f_count++ ], dta.name );
        print_fname( dta, f_count );
        flags = Fnext();
    }

/* Clear all remaining lines */
    for ( i = f_count + 1; i < MAXFILES; i++ )
    {
        locate( i * STARTLIN, 0 );
        clear_line();
    }

/* Some additional information */
    locate( 7, 0 );
    clear_line();
    printf( "Search string: %s\n", search );
    locate( 20, 0 );
    clear_line();
    printf( "Matches found: %d", f_count );
}

/* print fname() displays a particular file name */
void print_fname( dta_ptr, f )
struct dta_ptr *dta_ptr;
WORD f;
{
    locate( f * STARTLIN, 0 );
    clear_line();
    rev_on();
    printf( "%s", f );
    rev_off();
    locate( f * STARTLIN, 5 );
}

    printf( dta_ptr->name );
}

/* sev_err() displays a message about a severe error and exits */
void sev_err( str )
BYTE *str;
{
    locate( 23, 0 );
    clear_line();
    rev_on();
    printf( str );
    rev_off();
    Conout( BELL );
    getch();
    exit(1);
}

/* txt_center() uses the current value of width to place the
string pointed by 'str' in the center of the indicated line */
WORD txt_center( row, str )
WORD row;
BYTE *str;
{
    WORD len;

    len = strlen(str);
    if ( len > width )
        return( -1 );
    else
    {
        locate( row, (WORD) ((width-len)/2) );
        printf( str );
        return( 0 );
    }
}

WORD txt_right( row, str )
WORD row;
BYTE *str;
{
    WORD len;

    len = strlen( str );
    if ( len > width )
        return( -1 );
    else
    {
        locate( row, (width-len) );
        printf( str );
        return( 0 );
    }
}

/* bottom_line() puts additional instructions on line 23 */
void bottom_line()
{
    locate( 23, 0 );
    clear_line();
    printf( "RETURN -> Exit" );
    txt_right( 23, "HELP -> Search String" );
}

/* build_fname() constructs a file spec from a search string and file name */
void build_fname( search, newspec, fname )
BYTE *search, *newspec, *fname;
WORD i;
{
    strcpy( newspec, search );
    i = strlen( newspec ) - 1;
    while ( ( i ) && ( newspec[ i ] != '\\' ) && ( newspec[ i + 1 ] != ':' ) )
        if ( i )
            newspec[ i + 1 ] = '\\';
    else
        newspec[ 0 ] = '\\';
    strcat( newspec, fname );
}

```

Continued on page ==> 28



Bunky's EZ-Print

by Hugh McLean

Perhaps this article should be called "*A novices' guide to practical pascal*". As a non-professional programmer, I tend to be less concerned with "*twiddling bits*" and more concerned with results. Consider, that few people really think about how many degrees the timing on their automobile engine advances when they accelerate, or how much counterweight mass is required on the crankshaft. Most people just get in their car and go, being more absorbed with the traveling than the details of the mechanism. To many, the computer is like the automobile - merely a device to enable them to achieve a desired result with the least amount of '*hassle*'. Most of the formal information available for the ST is either aimed at the 'bit twiddler' that wants to know all the parameters of some obscure Gem function call--(in C of course)-- or directed at the '*User Only*' category.


I made a small, simple program with '*ALICE*', my favorite pascal interpreter, to control my printer output, and I thought it might be nice to have as a 'desk accessory'. This seemed like a good idea at the time, since I was always printing out something, like program doc files, and pascal (.pas) code that I wanted to keep for a reference in a binder or loose leaf notebook. I always had the problem of the print hitting exactly on the perforation line between the sheets, and the hole punch cutting out something I wanted to see. Another thing was that the type of binder that I use obscures about 1/2

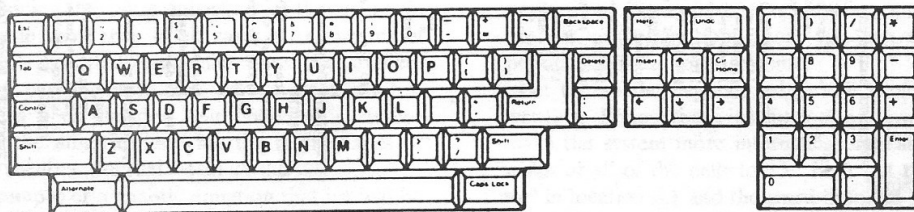
inch off the left margin of the page. If that wasn't enough, the ribbon on my old workhorse Gemini 10 is usually so worn that the print passes light. Sure, I could load in a word processor, and get nice text...but sometimes I only want a quick copy that doesn't have 15 font sizes, spoolers, buffers, dictionaries and thesauruses to load in. All I want is a crisp copy that I can read, and is good enough to fit in a notebook.

I guess the most difficult thing about the desk accessory program is finding the correct source code to work from. After much frustration with all the examples I had, and with no success using OSS Ver. 2.0, I wrote to OSS for more information. The manual suggests examining the source demo on the disk, but there wasn't any on my particular edition. ICD sent me the version 2.02, which had the sample demo, as well as a number of other corrections for irritating bugs. Part of my problem was that there was too much stuff in the sample that really wasn't used. This typifies programming GEM, (in my mind anyway), because so much of it is so 'all encompassing' that it is difficult to do something simple without addressing a lot of stuff you really don't need anyway. I miss the predefined variable '*KBD*:' that is a predefined file variable in *Turbo* and *Alice*, which I use quite often. Some examples will give you *FUNCTIONS* for *Menu_Register*, which is already declared as an *External Function* in version 2.02. The

This program was one of the most exciting things that I've done with my ST computer since I purchased it. For someone that struggles with programming code, this was an accomplishment to me that was its own reward, because I found this simple little desk accessory so valu-

Selecting 'Done' will configure the printer, and selecting Nevermind will reset the printer to the 'switch on' condition.

MORE 





Chaos With Your Computer

By Sol Guber

Despite the title and the inherent paradox of using a very deterministic machine, the micro computer has been a very useful tool to study **Chaos**. One of the major reasons that fractals and Mandelbrot spaces became so quickly accepted was the fact that computer programs could be written that helped visualize the underlying concepts. The computer and the proper program also allows the same sort of self discovery of **Chaos**, since a micro computer allows for the definition of equations and helps in the visualization of the results by use of colors and magnification schemes. What I would like to do is show you several examples of equations that turn chaotic using a spread sheet program like VIP Professional.

Before I start, I do need to explain that **Chaos** is starting to become a mathematical science. It uses the idea that a very small change in a factor can effect items all out of proportion to the change. It is almost as if the old proverb explaining the cause and effect of the want of a nail on a battle were true in all cases. As systems of equations that describe actual situations are studied, this condition became more and more evident, until at present, it has been decided that non-chaotic studies should be in the minority and that the mathematics of **Chaos** is the way that the real world works.

The first example of a chaotic equation that we will be studying is a very simple one. It was derived from

population ecology, to determine how the population of one generation will affect the population of the next generation. For example let us suppose that the dandelion population in an area, was dependent on the number of seeds that were produced, and that there was only one generation per year. Thus the higher the amount of seeds, the more dandelions were to be found. However, if there are many dandelion plumes around, the birds and insects might increase their utilization of them for food and the number of dandelions would decrease. The mathematical function that describes both of these equations is as follows:

next generation = factor * this generation * (1 - this generation).

Abbreviating next generation and this generation to ng and tg, the equation becomes

$$ng = f * tg * (1 - tg)$$

where both ng and tg are fractions of the total possible population. Thus if you assume a factor and set up the present dandelion population, you can determine the population in the years to come.

Let us use the capabilities of VIP Professional to do just this. Follow these instructions and away we go. To make the system more interesting, increase the column width of all of the cells to 15. First put the word 'Factor' in location A1 and the word 'Pop.' in B1. Enter the value 1.5 into A2 and the value .55 into B2. Using the

Factor = 1.5

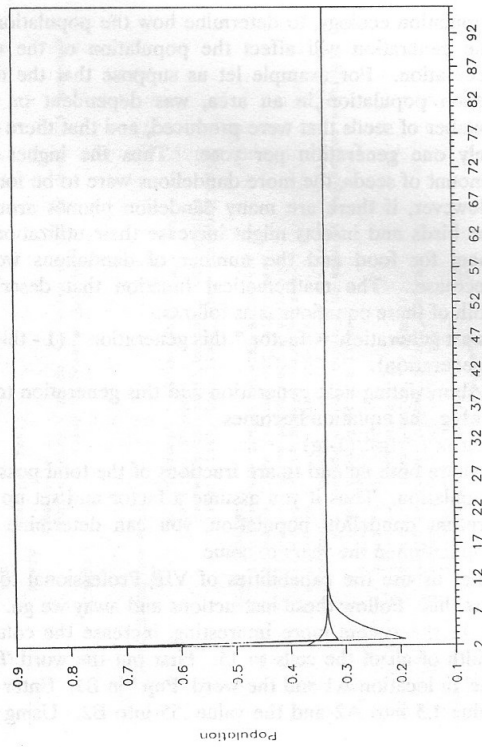


Fig. 1

Factor = 2.95

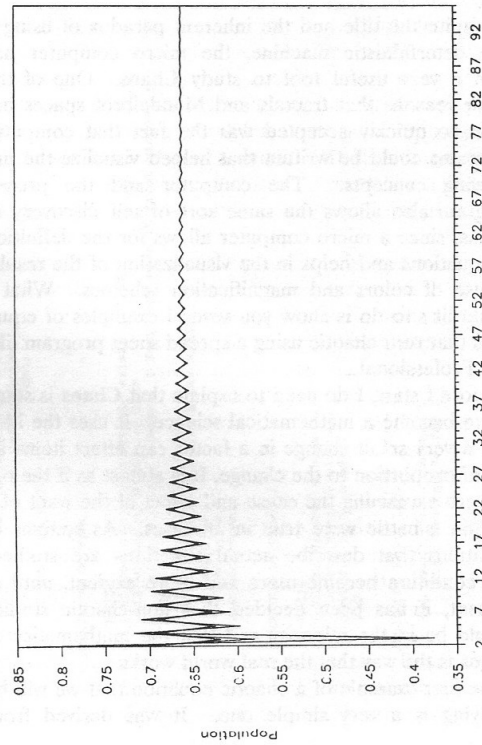


Fig. 3

Factor = 1.1

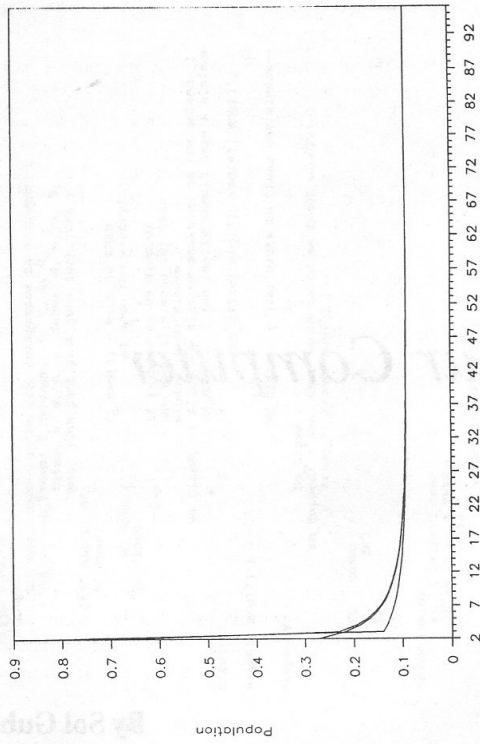


Fig. 2

Factor = 3.05

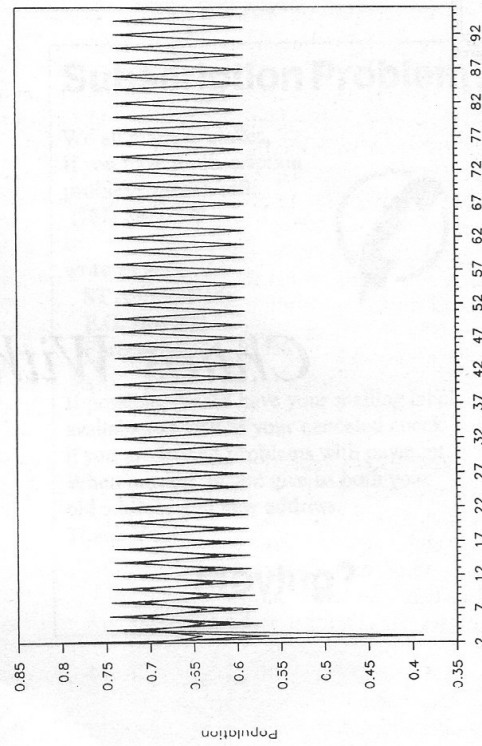


Fig. 4

DATA FILL function, put the numbers 2-100 into the A column starting at A3. Into B3, put the equation:
 $\$A\$2*B2*(1-B2)$.

(NOTE: Be sure to use the absolute address of location A2.) Using the copy function, copy B3..B3 into the cells B4..B103. Now using the copy function, copy the cells B1..B103 to C1..D103. Change the value of C2 to .70 and the value of D2 to .85. Now that we have the equation and several starting values into the system, set up the graphing function.

Set up the program so that the graph type is Line with the X variable being columns A3..A103 and the A, B, and C variables being B3..B103, C3..C103, and D3..D103 respectively. To make the graph easier to read, change the options so that only every fifth point is shown. Finally have the graphs be shown in color. Now we can play with it.

If you press F10 to show the graph, you will see three lines converging to about .37 and the graph seems to be quite stable. No matter what the initial population is, the final value seems to be the same. (If you remember, the values were .55, .70, and .85 for the initial population densities. Since we have the factor (1-tg) in the equation we do not need to have any population densities below .50). Press any key and change A2, the factor from 1.5 to 1.6. Press F10 to view the graph and you will see that there is not much difference to the curve, but that it stabilized at a slightly higher value. Change A2 to 1.1 and the graph now shows that the dandelion population will drop to about 0.1 in several years and again remain fairly constant.

Thus far, this seems to be a very stable system and if you believe in the equation, it might be useful to predict populations. If you are really careful and wish to try you hand at optimizing the dandelion population, you can determine that if the factor is above 1.145 then the population will reach a stable plateau and if it is below 1.145, then the dandelions will die out after a period of time.

Why have I picked this equation to show chaotic behavior? It seems to be very stable. OK, let us change the factor some more and see what happens. This time, besides looking at the graph, let us determine how fast the population converges. For our definition of convergence, let us use three decimal place as being equivalent. Thus there are three similar numbers next to the decimal, then the values are the same. Slowly change the factor from 1.1 to 3.0. Every so often, press F10 to see what the graph looks like.

Table 1 shows the results that I have generated. As you increase the values of the factor from 1.1 to 2, the number of generations it takes for the value to reach a plateau seems to decrease from about 45 to about 6. As you increase past 2.0, the number of generations slowly increase until a factor value of 2.8 and then it quickly increase at a factor of 3.0. Change the value to 3.0 and look at the curve again and as a comparison check the

values at 2.9, 2.95, 2.99, 3.01, 3.05 and 3.1. **Figure 2** shows what I have seen.

If you change the values of the factor around the range of 3.0, a very strange behavior appears in the graphs. If you look, the perturbations seem to increase and after 3.0, the population values do not look as if they will approach a single value. This is true, after 3.0, the function no longer converges. There seems to be two populations that alternate. In the mathematical theory that we have been using, this means that if the factor is greater than 3.0, the population will alternate between two values. What is even stranger is that the three initial values all seem to converge to the same two values of the population.

What is the difference between a factor of 2.99 and a factor of 3.01 in our model? The actions of the graph did seem to increase gradually and there did not seem to be any reason that the convergence did not go to a constant value. This is the whole idea behind **Chaos**, that a very small modification in one value will cause large changes in the graph. However, the fun is just beginning.

Thus far, there does not seem to be an effect due to the initial values of the population. Let the value of the factor increase slowly and look at the graphs that are resulting. All of a sudden at a factor of 3.4, there seems to be a difference between the initial values of the population. The final values after many generations seem to be the same, but the actual values seem to alternate generations.

Let the factor increase even further. At about 3.5, it looks as if the population densities have now split into four numbers that seem to alternate. Also it seems that the initial population value is not critical again. As you keep typing in factors and approaching 4.0, which is the limit for this function, the graph changes more and more. It does not seem that there is a pattern any more and that the initial population value is highly critical. At about 4.6, there seems to be about 8 values that shift from one to another and that the values seem to be highly critical of the initial value.

This does not seem to be very logical. During much of the region for factors between 1.0 and 3.0 there does not seem to be any difference between the values that the factor has and the initial population density values. After 3.0 the final convergence does not occur and there is so much difference in the appearance of the graph as both the factor changes and the initial is modified.

Let us look at another equation and perform the same type of analysis. This time the equation will be $X_n = r * \sin(\pi * X_{n-1})$. Set up the spread sheet as follows: into A1 type "Factor" and into A2 put 1.0. Using the **Data-Fill** function fill in from A3..a100 with the numbers 2 through 100. Into B1 put pop and into B2 put .6. Into C1, put pop2. Into B3 put the equation:

$\$A\$2 * @SIN(@PI * B2)$.

MORE 

Factor = 3.50

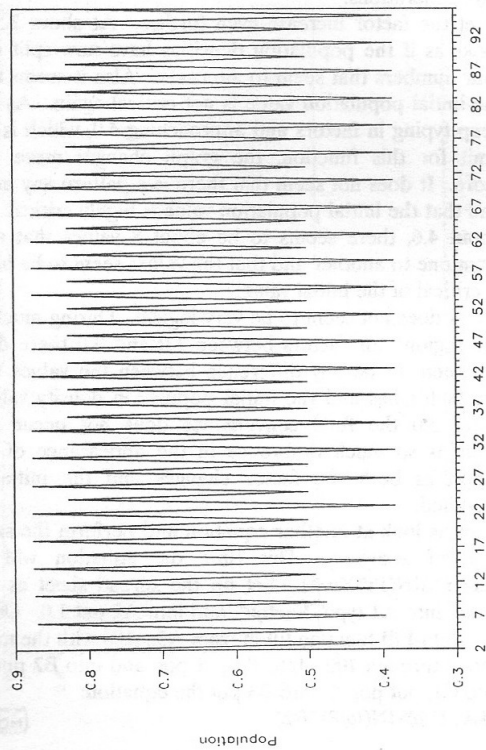


Fig. 5

Factor = 3.95

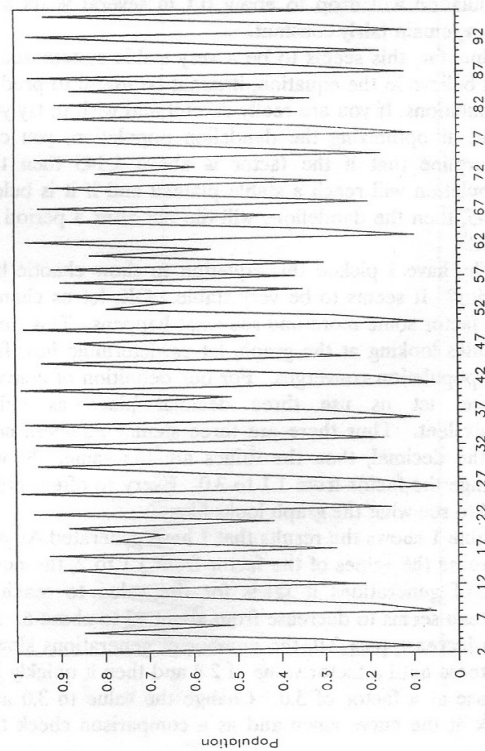


Fig. 7

Factor = 3.70

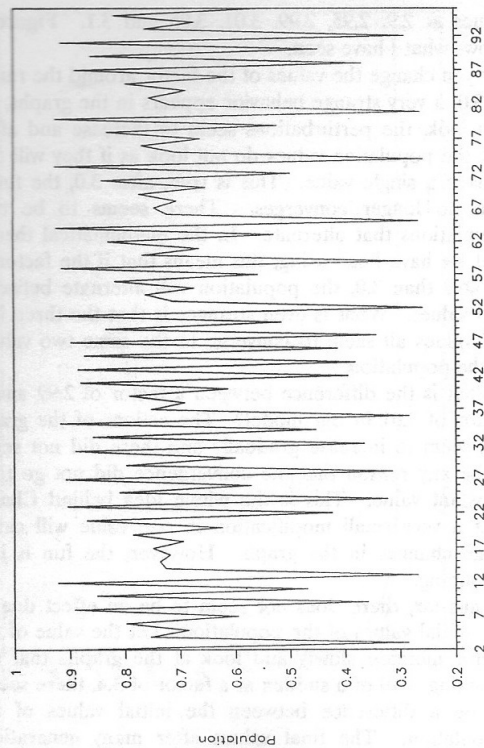


Fig. 6

Cluster at Factor = 1.3

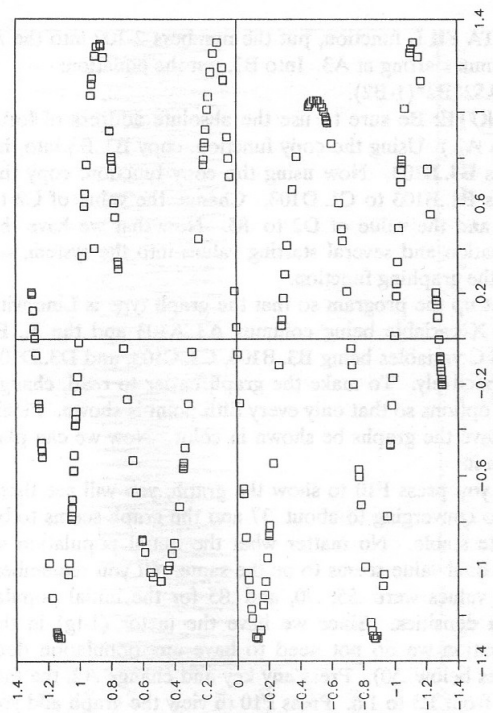


Fig. 8

Now copy this equation from B3..B3 into B4..B100. Now into C3 put B2 and copy from C3..C3 to C4..C100. The graphing will be the same in the first example, as was performed the last time, with a simple Line graph without symbols using the A3..A100 column for the X variable and B3..B100 for the A. variable. Press F10 to view the graph and you will see that it is just as chaotic as some of the other graphs that we have seen.

Let the value of the factor vary from the initial value of 1.0 and gradually change it to 2.0, by about .33. Examine the graphs that are occurring and you will see that the graph is very similar to others. However, this example is one that is known as an attractor and I tried to trick you with this example. If you use the value of 1.5 for the factor and examine the graph, you will notice that the graph converges and converges very quickly to either of two values. If you move slightly from the values of 1.5, it becomes chaotic again. There are many places along the number line where all of a sudden, the system dies down and becomes calm. At any of the values that end in .5, the values converge either to a single value or to a multiple value.

There is a second way to look at this equation. Go to graph and press Reset. Now go to Type and press XY. Use B3..B100 for the X range and C3..C100 for the Y range. View the graph at some point where the graph is 'chaotic' and all of a sudden, it seems to form a pattern. An example can be seen in Fig. 8. The graph is similar to a sine curve laid on its side. If you use a value of 1.5 for the factor, the graph seems to disappear and there is just a group of points. Move slightly away from the

value of 1.5 and the graph reappears. The factor can be made larger for the factor, as long as there is no approach to a .5 value, and the graph appears that seems to have a very significant value. Again, it is very fun to explore this equation because there are many subtle attractors around that causes the graph to disappear. An example of a subtle attractor is about 2.74 (Fig. 9), where the 'Chaos' disappears into two little sine type waves.

These two example are just the beginning of what can be studied in the field of **Chaos**. VIP Professional can be used as a tool for exploring the behavior of equations that are both interesting and have relationships that vary very subtly. **Chaos** is a science of the future that can be examined today with many fine tools.

Before I leave the subject, I would like to quote from James Gleick who has brought **Chaos** to the masses, from page 311: "As a growing snow flake falls to earth, typically floating in the wind for an hour or more, the choices made by the branching tips at any instant depend sensitively on such things as the temperature, the humidity, and the presence of impurities in the atmosphere. The six tips of a single snowflake, spreading within a millimeter space, feel the same temperature and because the laws of growth are purely deterministic, they maintain a near perfect symmetry. But the nature of turbulent air is such that any pair of snowflakes will experience very different paths. The final flake records the history of the changing weather conditions it has experienced, and the combinations may as well be infinite.

FactorConvergence

1.145
1.319
1.610
1.8 6
2.0 6
2.1 6
2.211
2.615
2.825
2.945
3.0300

Table 1

2.90Stable
2.95Stable
2.99The perturbation are definitely decreasing. the different colors for the three vales are visible.
3.00Different colors visible
3.01Different colors visible, the perturbation is stable.
3.05Stable perturbation, Single color visible
3.10Stable perturbation, single color visible after 30 generations

Table 2

Cluster Point at Factor = 1.74

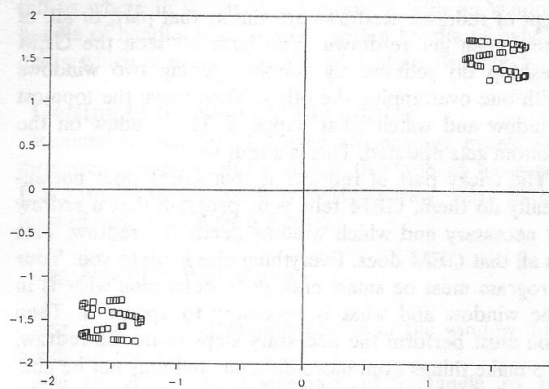


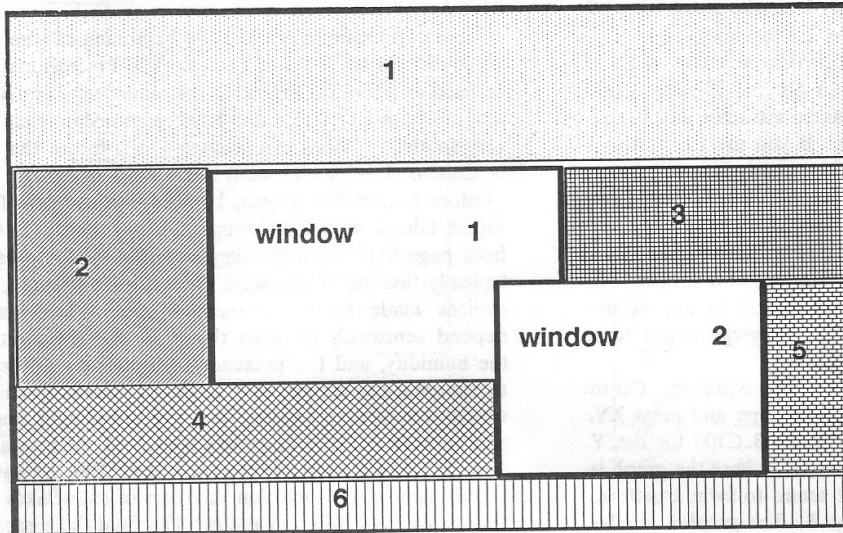
Fig. 9

END



Overlapping Windows and Redraws

in
Pascal



Rectangle List

Figure 1

by Bruce Wiebe

When I think of GEM, I think of windows. One cannot fully understand windows until you master the concept of redraws. Redraws are simply that part, or all, of the screen get re-drawn. You have all seen the GEM desktop do redraws by simply opening two windows with one overlapping the other. Then move the topmost window and watch what happens. The window on the bottom gets updated. This is a redraw.

The tricky part of redraws is that GEM does not actually do them. GEM tells your program that a redraw is necessary and which window needs the redraw. This is all that GEM does. Everything else is up to you. Your program must be smart enough to determine what is in the window and what is necessary to update it. Then you must perform the necessary steps to do the redraw. To make things even more difficult, you may not be able to redraw the complete window as part of it may be obscured by another window. You would not want to over-write another window.

Before I go ahead and explain how to do redraws, you should be aware of what causes a redraw. (The example program illustrates this as well as how to do the redraws.) Basically, whenever an action takes place that demands that more of the window be shown than already is, this causes a redraw message to be issued by GEM.

This includes:

- 1) a window being moved from over top another window
- 2) a window which was partially off the screen is moved more towards the centre of the screen.
- 3) a window is made bigger via the size box. Note that making a window smaller does not cause a redraw message.
- 4) a bottom window is clicked on which tells GEM to make this the top window.
- 5) a top window is closed which leaves a window underneath.

All five of the above examples cause GEM to issue the redraw message to your application program. But how do you get the message? You have to set up your program to look for these messages. This is easy with the `Get_Event` function.

`Get_Event` is a function that will monitor all portions of GEM and tell you what the user did. It is usually advisable to only have one such call to `Get_Event` in your program. The function has many parameters and looks

complicated to use. Fortunately, I am only using 2 out of the 22 possible parameters. The rest are just junk.

```
CONST
  E_Message = $01;
VAR
  msg_area : Message_Buffer;
  event,junk : Short_Integer;

event:=Get_Event(E_Message,0,0,0,0,FALSE,0,0,0,0,
FALSE,0,0,0,0,msg_area,junk,junk,junk,junk,junk);
```

Parameter 1 is a bit-mapped value telling Get_Event what kind of events to look for. A GEM message is an event and is the only event I am interested in. Parameter 16 is a variable of type Message_Buffer (I'm using msg_area) which holds the message information that GEM returns via the Get_Event function.

The variable event contains the type of event returned, in this case it will always be an event message since this is only type of message I am checking for. Even so, you should always check and make sure that event really does contain this value. This is because you might want to allow for another type of event without explicitly checking for it, namely the timeout message. Once you have determined that you are dealing with a message event, the next task is to find out which one.

The possible message events are:

- 1) a redraw message (WM_Redraw)
- 2) a request to resize or move a window (WM_Size, WM_Moved)
- 3) a request to bring a background window to the front (WM_Topped)
- 4) a menu selection (MN_Selected)

WM_Size and WM_Moved are handled by the same routine and are very simple. See the program listing for an explanation.

WM_Topped is kind of interesting. All you have to do here is issue a call to Bring_To_Front with the appropriate parameters. But do you not have to update the window? No, GEM will issue the redraw message for you and your program can handle it as it does any other redraw message.

By now you should probably run the example program. Double-click on REDRAW.PRG. For each of the four windows an alert box will come up asking you what type of windows you want, text or graphics (see figure 2). Mix them up to make things more interesting. Since all windows open full size, you will have to resize them to see what is going on. Feel free to play around and see what is going on. Experiment with the above mentioned ideas by clicking in a background window, moving and

resizing windows, etc. When running, the program will look something like figure 1. Note the overlapping windows. Each window can have different information.

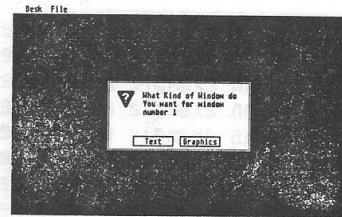


Figure 2

Your program must be able to distinguish between the various windows and know how to recreate each one. I find that by storing window information in an array of records, the job is almost done.

Here is the window structure:

```
TYPE
  Kindof_Window = (Textbased,Graphics);

Window_Type = Record
  handle      : Integer; {GEM's window handle}
  Type_of_Window : Kind_of_Window;
  windowTitle  : Window_Title;
End;
VAR
  Windows:Array[1..Max_Windows] OF Window_Type;
```

All the information we want is in the record. We also need another function called which_window which will tell us where in the array a window is with an window handle of handle. If no window with a handle of handle exists, which_window returns a value of no_window.

Getting to the heart of the matter

Remember that GEM passes all message events via msg_area. msg_area[0] contains the type of message. For now, let us assume it is WM_Redraw. Here is the form of msg_area in such a case:

```
msg_area[0] = WM_Redraw

msg_area[3] = the window handle of the window that
needs the redraw
msg_area[4] = x1 coordinate of rectangle to be
redrawn
msg_area[5] = y1 coordinate of rectangle to be
redrawn
msg_area[6] = x2 coordinate of rectangle to be
redrawn
msg_area[7] = y2 coordinate of rectangle to be
redrawn
```

MORE

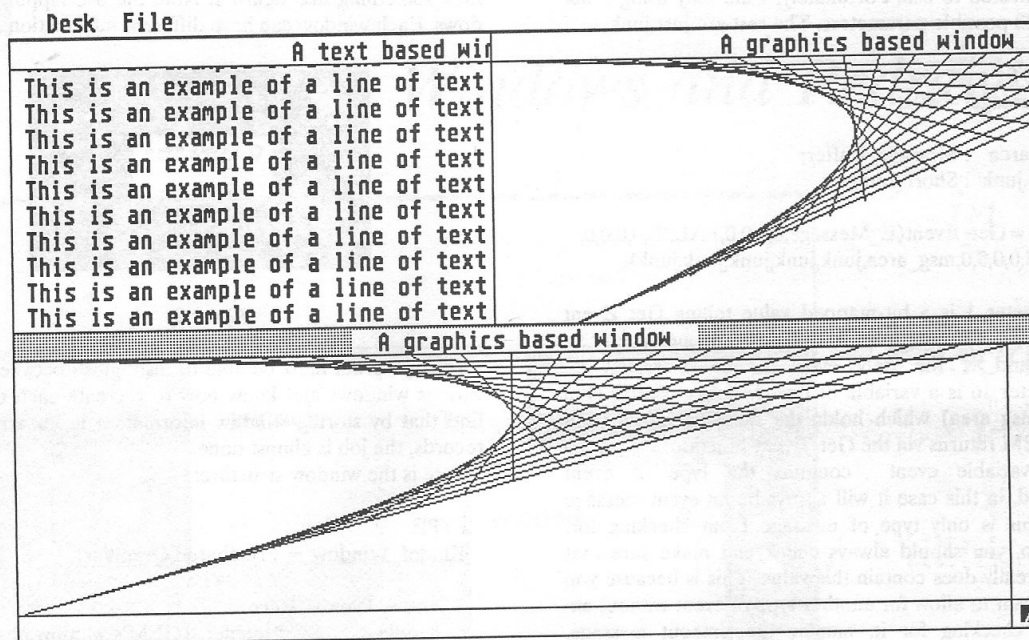


Figure 3

What are all these rectangles about? Well, the screen is updated one rectangle at a time. GEM uses an internal algorithm to divide the screen into the minimum number of non-overlapping rectangles necessary to cover the screen. Note that if you have two or more windows on the screen, it may take several rectangles to do this. (See figure 3)

Personal Pascal provides three calls that use these rectangles and determine if they need to be redrawn. These calls are:

```
First_Rect
Next_Rect
Rect_Intersect
```

Look at the procedure `Do_Redraw` in the example program. It is passed five parameters, the window handle, and the four coordinates of the rectangle that needs the redraw. As is the case with any redraw, the first and last call are to `Begin_Update` and `End_Update`, respectively. This ensures that nothing will change the screen while the redraw is going on.

Then I call `First_Rect`. This returns the first rectangle in the rectangle list discussed above. Inside the `While` loop, I make the call to `Rect_Intersect`. This takes eight parameters. The first four are the coordinates of the window needing the redraw; the next four are the coordinates of the rectangle returned by `First_Rect`. If these

two rectangles intersect, `Rect_Intersect` returns `TRUE`, meaning that the rectangle needs redrawing.

I do this in a simple way. I set the clipping to the size of the rectangle and then redraw the entire window. Although this always works, you may want to try a faster method.

Continuing in the loop, I make a call to `Next_Rect` which returns the coordinates of the next rectangle in the rectangle list. Another call to `Rect_Intersect` is made and the program continues on. The loop is ended when the width or height of the returned rectangle is zero.

Now look at the procedure `Redraw_Window`. The only thing tricky here is to determine what to actually put in the window. Since I saved information as to whether the window is a text or graphics window, this job is simplified. But what if we have many windows with all totally different contents? It is still up to you to make sure that your program can determine what is in each window and be able to recreate them at any time.

As you can see, handling windows and redraws is not all that complicated. Make sure you know what is in each window and you should not have any problems. You will be much more impressed with your own windows than those written by others. Trust me!

```

Program Redraw;
($I Gensubs.pas)
($I Auxsubs.pas)
CONST
  Max_Windows = 4;
ExampleText =
  'This is an example of a line of text to go into a
  text-based window.';
TYPE
  KindOf_Window = (Textbased,Graphics);
  Window_Type = Record
    handle      : Integer;
    TypeOf_Window : KindOf_Window;
    WindowTitle : Window_Title;
  End;
  window_range = no_window..Max_Windows;
VAR
  (window items)
  x,y,w,h,
  junk,
  CharW,CharH : Integer;
  Window_Parts : Short_Integer;
  msg_area : Message_Buffer;
  Windows : Array[1..Max_Windows] OF Window_Type;
  (menu items)
  menu_bar : Menu_ptr;
  FileMn,
  QuitMn : Short_Integer;
  (misc items)
  event : Short_Integer;
  done : Boolean;
FUNCTION which_window( wind_handle: integer ): window_range;
VAR
  i : integer; { Used to index through the 'windows'
  array }
  found : boolean; { Used to indicate when we find a
  matching window }
BEGIN
  i := 1;
  found := false;
  WHILE (i <= max_windows) AND NOT found DO
    { If we find a matching window, then set our exit flag }
    IF Windows[i].handle = wind_handle THEN
      found := true
    ELSE
      i := i + 1;
  IF found THEN
    which_window := i
  ELSE
    which_window := No_Window
  END;
PROCEDURE Construct_Menu_Bar;
BEGIN (construct menu_bar)
  menu_bar := New_Menu(20, 'About Redraw...');
  FileMn := Add_Menu(menu_bar, 'File');
  QuitMn := Add_Menu(menu_bar, 'Quit');
  Draw_Menu(menu_bar);
END; (Construct menu_bar)
PROCEDURE Redraw_Window(window,x,y,w,h : Integer);
{clipping must already be set for this routine to work}
VAR
  LineCnt,i,
  Bottom_Height,
  x1,y1,w1,h1 : Integer;
  window_range;

```

```

  Begin
    Hide_Mouse;
    Paint_Color(White);
    Paint_Rect(x,y,w,h); (clean up display window)
    WindowIndex:=Which_Window(window);
    With Windows[Index] Do
      Begin
        Work_Rect(window,x1,y1,w1,h1);
        Height:=h1 DIV CharH;
        Construct_Window_OF_Textbased( text based window)
      Begin
        For LineCnt:=1 To Height DO
          Begin
            Draw_String(CharW*W1,WhereY+y1,ExampleText);
            WhereY:=WhereY+CharH;
          End;
        End;
      End;
    End;
  End;
  Graphics: (Draw Walking Line)
  For i:= 1 to 20 Do
    Line(x1+i*(w1 DIV 20),y1,x1+i*(w1 DIV 20),y1+i*(h1 DIV 20));
    (This is just a neat looking, simple graphic)
  End; (case)
  End; (with)
  Show_Mouse;
End; (Redraw_Window)

PROCEDURE Do_Size;
VAR
  msg_area[3] : window_handle
  [4]
  [5]
  [6]
  [7] : coordinates of window to be sized)
  Set_Wsize(msg_area[3],msg_area[4],msg_area[5],
  msg_area[6],msg_area[7]);
End; (Do_Size)

PROCEDURE Do_Redraw(window,x0,y0,w0,h0 : Integer);
BEGIN
  Paint_Color(White);
  Begin_Update;
  Hide_Mouse;
  First_Rect(Window,x,y,w,h); (get first rect of GEN rect
  list)
  While (w<>0) OR (h<>0) DO
    Begin
      If Rect_Intersect(x0,y0,w0,h0,x,y,w,h) THEN
        Set_Clip(x,y,w,h);
        Paint_Rect(x,y,w,h);
        Redraw_Window(window,x,y,w,h);
      End;
      Next_Rect(window,x,y,w,h); ('walk' the rectangle list)
    End;
  Show_Mouse;
  Set_Mouse(M_Arrow);
  End_Update;
End; (Do_Redraw)

PROCEDURE Do_Menu_Selected;

```



```

1: Begin
  Typeof Window:=textbased;
  windowtitle:= ' A text based window ' ;
End;

2: Begin
  Typeof Window:=Graphics;
  windowtitle:= ' A graphics based window ' ;
End;

End; (case)
Handle:=New_Window(window_parts,windowtitle,0,0,0,0);
Open_Window(Handle,0,0,0);
End; (for)
End; (Initialize)
Begin (main program starts here)
  If Init_Gem < 0 then Halt;
  Initialize;
  Repeat
    Event:=Get_Event(E_Message,0,0,0,FALSE,0,0,0,0,
      FALSE,0,0,0,0,msg_area,junk,
      junk,junk,junk,junk);
  Case event Of
    E_Message: Do_Event_Message;
  Until Done;
  Exit_Prog;
End.

```

END

More Desktop Selector from page 15

```

/* locate() positions the cursor at a particular row and column */
void locate( row, col )
short int row, col;
{
  Cconout( ESC );          Cconout('Y');
  Cconout( row + 32 );     Cconout( col + 32 );
}

/* clear_line() clears the rest of the current line */
void clear_line()
{
  Cconout( ESC );         Cconout( 'K' );
}

/* rev_on() enables the reverse attribute of on-screen attributes */
void rev_on()
{
  Cconout( ESC );        Cconout( 'p' );
}

/* rev_off() disables the reverse attribute */
void rev_off()
{
  Cconout( ESC );        Cconout( 'q' );
}

```

END

```

Procedure Desk_MN;
Begin
  junk:=Do_Alert
  ('[O]This is just a simple example of how redraws work. [by
  Bruce Wiebe][OK]');
  1 Menu_Normal(Menu_Bar, 3);
  End;

Procedure File_MN;
Begin
  If msg_area[4] = QuitIN Then done:=TRUE;
  Menu_Normal(Menu_Bar,FileMN);
  End;

Begin (Do Menu Selected)
  Desk_MN := Do Menu Desk MN;
  If msg_area[3] = FileMN then File_MN;
  End; (Do_Menu_Selected)

Procedure Do_Event_Message;
Begin (Do_Event_Message)
  Case msg_area[0] Of
    Do_redraw(msg_area[3],msg_area[4],msg_area[5],
      msg_area[6],msg_area[7]);
    WM_Sized,WM_Moved : Do_Size;
    WM_Topped         : Bring_To_Front(msg_area[3]);
    WM_Selected: Do_Menu_Selected;
  End; (case)
  End; (Do_Event_Message)

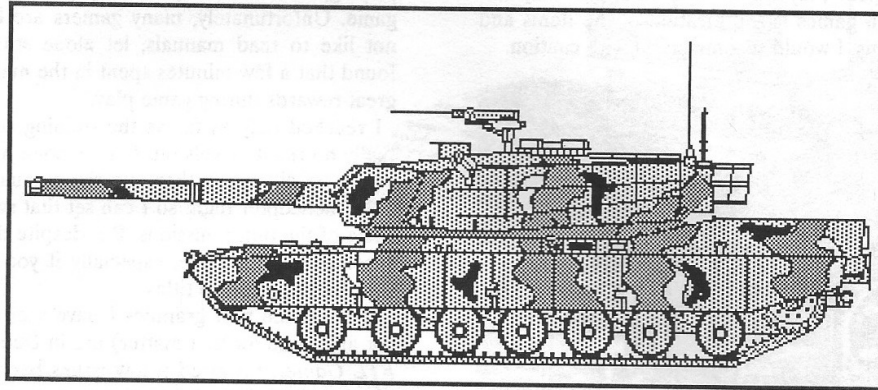
Procedure Exit_Prog;
VAR
  number : Integer;
Begin
  number :=1 to Max_Windows DO
  For Which_Window(number) <> No_Window then
  Begin
    Close_Window(number);
    Delete_Window(number);
  End;
  Erase_Menu(Menu_Bar);
  Delete_Menu(Menu_Bar);
  Exit_Gem;
End;

Procedure Initialize;
VAR
  i,result : Integer;
  prompt : String(255);
Begin
  construct_menu_bar; Check junk,junk;
  sysfont_size:=CharW; CharH junk,junk;
  sysfont_size:=CharW; CharH junk,junk;
  For i:=1 to Max_Windows do
  Begin
    With Windows[i] Do
    Begin
      Prompt:=Concat(
        CHR(i+$10),What Kind of Window do You want for window number
        i+$10);
      Prompt:=Concat(Prompt, '[Text|Graphics]');
      result:=Do_Alert(Prompt,0);
      Case result Of

```



The GameMaSTer



by R. Bradley Andrews

Welcome to another month of your favorite column. It is, isn't it? Of course it is... I thought so. This month I plan on continuing to chop away at the backlog that has accrued due to some publication problems.

The good news for ST owners is that it appears that there is finally a large number of game software titles. Unfortunately, among many game companies the ST is not viewed as having much of a future. Please buy as many of the games that appeal to you as possible so we can continue having a large selection of games for the ST.

First on the list this month is *Slaygon* by *Microdeal*. Unlike many of the other *Microdeal* offerings, *Slaygon* is more of a strategy game than a shoot-em-up. In it you take control of a super-robot and use it to infiltrate a dangerous enemy lab with the goal being to cause the lab's reactor to overload, blow up, and destroy the lab. You must also get out alive prior to the explosion, although this is the least of your worries. The game is entirely mouse driven, which can be a drawback when you would like to use the arrow keys on the keyboard for movement, but on the whole it works out reasonably well.

As with other games from *Microdeal/Michtron*, the documentation is short but enough information is included to start out. Definitely read the whole manual

prior to play, unless you are one of those people who like mental anguish.

A couple of hints for those just starting in the game: pressing the "???" icon when an object is in front of your robot will tell you the name of that device. You can then avoid a lot of the guessing that characterized my first several attempts at the game.

Also, installing a new weapon or shield on your robot causes the old one to take its place in your inventory. For example, if you are currently using a green shield and then select and install a blue one, the green one will now occupy the same slot in your inventory. (The same icon will be there, but its color will have changed from blue to green.)

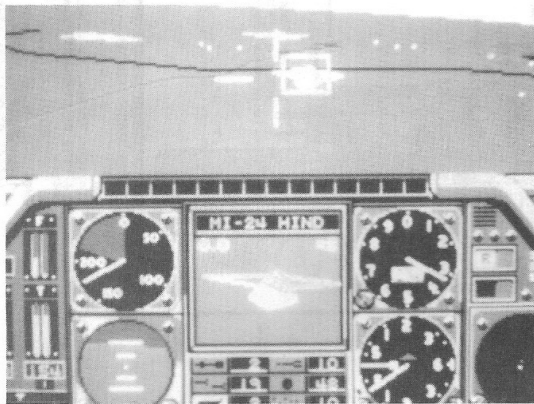
Games can be saved between sessions, which is a plus, especially given the potential for a lengthy game. It is advantageous to play in one sitting if possible, or to at least take some notes so you can remember where the important locations are around the lab.

Each game is set up randomly based on the level of difficulty you chose. Higher levels feature more randomness and it may be impossible to win a game at times because of an important object is unreachable, such as the key that opens all orange doors may itself be hidden behind an orange door. This problem does not occur at lower levels, and only rarely at the higher ones. The

manual does warn about this and other potential problems.

Your robot includes a built-in mapping capability, which is very helpful. But, as with every other action in this game, it uses up some of your valuable energy. Energy can be recharged by either the recharge items lying around in some of the areas or in one of the power rooms, provided you have the rod that allows you to tap into the energy source there.

On the whole, it was an enjoyable game, I would love to find out how they did the graphics. It may not stand up to many repeat playings though, since the only difference between games is the location of the items and the special rooms. I would recommend it with caution.



Gunship

Gunship by *Microprose* is a very detailed simulation of an AH-64 Apache helicopter. As with *F-15 Strike Eagle*, they have once again provided gamers with a very detailed simulation of an interesting area of life. The 3-D graphics and the spectrum of options combine to provide for a very good simulation.

Several areas of the world are available for missions. These are: a training range in the USA, Southeast Asia, Central America, the Middle East, and Western Europe. Flying style also can vary between Regular Missions, easiest style; the standard, Volunteer Missions: a little tougher; and Volunteer Hazardous Duty: the toughest of all. Obviously, the more dangerous the mission, the better any medals are that you get after a successful outcome. The game also features several "reality" settings, thus allowing for even more customizing of the adventure.

Prior to flying a mission, you are briefed on the mission and you have an opportunity to go on "sick call" to avoid the mission if you feel that it is too dangerous. If you go ahead with the mission, you have the chance to load out the plane with the combination of fuel and weapons that you wish for the mission. Then your mission begins.

Play features the view out your view port, a "heads up" display, and other informative displays. Your radar sys-

tem can identify any enemies within your targeting area. You also have a large scale map that tells you everyone's approximate location. But you must be cautious when viewing it because your helicopter will still be flying. You can still be shot at or crash into a mountain.

The manual is over 70 pages long and it is vital to at least skim the first section prior to playing. This brings me to what I see as the biggest problem of the game. As with other *Microprose* titles, the amount of detail in the simulation requires the user to read the manual before playing, and an in-depth study prior to mastering the game. Unfortunately, many gamers are like me and do not like to read manuals, let alone study them. But I found that a few minutes spent in the manual resulted in great rewards during game play.

I reached only as far as the training session, with virtually no reality levels on. If I get some more time later, I plan on digging farther into the manual to understand "true" helicopter flight so I can set that switch on and try some of the other missions. But despite this, I still highly recommend this game, especially if you enjoyed any of the other *Microprose* titles.

Some of the best graphics I have seen in an ST game (or any game for that matter) are in **Dungeon Master** by *FTL Games*, reviewed a few issues back. You must explore and go down into a dungeon, one level at a time, destroy the bad guy, and free the universe, or something like that. However, the 3D graphics definitely make up for the predictability of the plot. Another interesting feature is that the game moves in real-time. If you do not move, the game continues on, monsters move, you get hungrier and thirstier, etc. Many puzzles also must be surmounted prior to finishing the quest.

The only real complaint is that there is a sparsity of food, especially before you get to the point where you can find and kill purple worms easily. (These produce edible purple worm slices when killed.) The game was all the rage on our local BBS for several weeks. Supposedly, *FTL* is planning a mini-adventure expansion disk that will allow you to use the same characters in a different dungeon.

I would like to generate some feedback as to the future of the ST, especially in the entertainment area. Can we expect to see more releases, and perhaps a few more new and innovative designs, a la **Dungeon Master**, to be brought out? Or will the growing opinion that the ST is a "dead machine" prevail?

As I stated at the start of the column, a fair number of good quality games are actually becoming available for the ST. But how long will this last? Can anything be done, within reason, to "turn the tide" as it were, or is the battle lost?

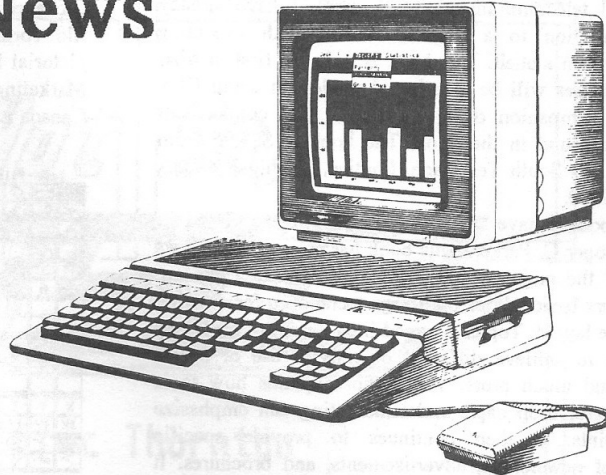
If you will send your opinions to me, either in care of **ST Applications** or on the BBS I mentioned in the June column, I will discuss them in a future issue.

Until my next column keep those aliens at bay...

(END)



LateST News



by Stephen Mortimer

Starglider II and Black Lamp

Starglider II and **Black Lamp** are brought to the US via Rainbird Software, formerly Firebird Licensees. Starglider II, the sequel to Starglider, features high-speed animation with solid, three dimensional graphics. The player controls the Icarus, a highly versatile spacecraft that has a stardrive warp engine, cargo bay, tractor beam, energy absorbing shields, and the capacity to add new weapon systems. All these devices will aid you in your quest to destroy the Egron Space Station and save the planet Novenia. In order to accomplish this mission, you must travel from planet to planet, searching for the weapons you need, while avoiding space pirates and collecting valuable cargo. The game has hundreds of different enemies and objects that you must learn to distinguish. Gameplay is fast and furious and much more enjoyable than the original Starglider, which is still quite good. The added dimension of outerspace takes away boredom of seeing the same terrain over and over. Some planets have harsh terrain with volcanic activity, while others contain a heavy enemy presence. Starglider II allows games to be saved in progress. Either a joystick or mouse can be used to control the Icarus. This is one of the few games that have held my attention in recent months. The combination of smooth flight, great graphics and music, and a shoot 'em up with a touch of adventure gives Rainbird a winner. It works on both color and monochrome systems, including the MEGA. A cassette of the theme song, poster, and a novella are included in the package. Password protection with words from the novella is used in this game. \$44.95 from Rainbird Software, 3885

Bohannon Drive, Menlo Park, California 94025. (415)322-0412.

Black Lamp is a whimsical adventure with cartoon quality characters. Impressive cartoon-like graphics and quality animation are the main selling points for **Black Lamp**, a game where the baddies are always attacking. The hero, Jolly Jack the jester, is out to capture the black lamp that is protected by a fire-breathing dragon. Once the lamp is returned to its proper resting place, the land will be free from evil. Jolly Jack has several lives in which to accomplish his quest. These will be sorely needed, since Jack's life force is quickly depleted by flying bats, wizards, and other forces of evil. These enemies can be destroyed by Jack's magic bolts. Magic bonuses can be obtained by collecting bonus objects throughout the game. These bonuses give you magic armor, super fire power, and the ability to jump off high places for a limited time. After playing this game several hours I found this game fun to play, yet very frustrating. There are just too many enemies to cope with. Several other people who have played this game, expressed mixed opinions about it. For \$24.95 retail, it is not that expensive to buy, but I would recommend trying it out at your dealer first. See above address.

GFA BASIC and DTP Book

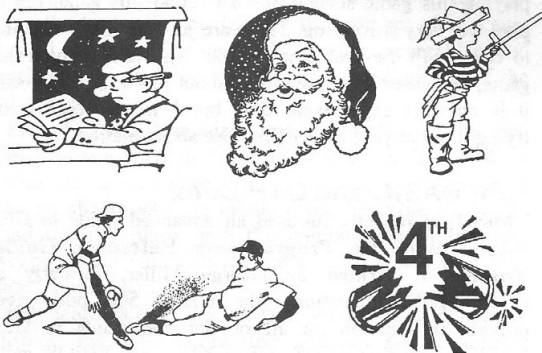
MichTron recently released an advanced guide to GFA BASIC called the **Programmers Reference Guide, Volume I**. Written by George Miller, formerly of COMPUTE! Publications, the book's 500 odd pages provide information on all of the commands in GFA BASIC, including BIOS, GEMDOS, and XBIOS calls.

Also included are several utility programs that illustrate many commands in context of an application. Seeing a command in this manner is much more beneficial than a mere description. The three main sample programs are a shape editor for animation sequences, a sound editor, and a full-featured telecommunications program. I have already found a solution to a problem I had with executing programs within a shell. This book is only the first in what the author hopes will be a series of guides on using GFA BASIC. A companion disk is available that contains all source code found in the book. The book is \$29.95 from MichTron, 576 South Telegraph, Pontiac, Michigan 48053. (313) 334-5700.

Another book I have come across is **Looking Good in Print** by Roger C. Parker. This book has seen good reviews in many of the major publishing magazines and I can see why. It offers basic advice on design techniques for creating a good page layout. Topics covered include use of fonts and type styles to achieve different objectives, use of space, graphics, and much more. The author explains how tools such as boxes, drop caps, and silhouetting can emphasize certain points. It then continues to provide specific examples of newsletters, advertisements, and brochures. It also discusses common pitfalls in design. **Looking Good in Print** offers sound advice on almost every topic in desktop publishing, yet it still remains a book suitable for all computers and applications. I highly recommend it to anyone who produces documents with desktop publishing programs on a regular basis. **Looking Good in Print** is \$23.95 from Ventana Press, Inc., PO Box 2468, Chapel Hill, North Carolina 27515. (919) 490-0062.

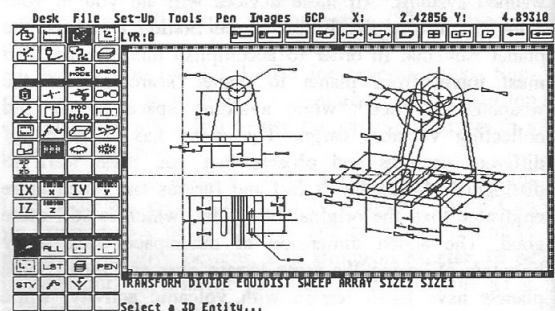
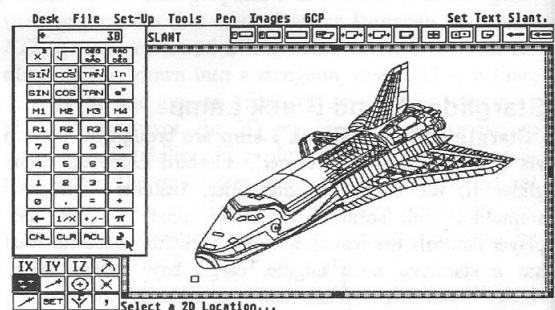
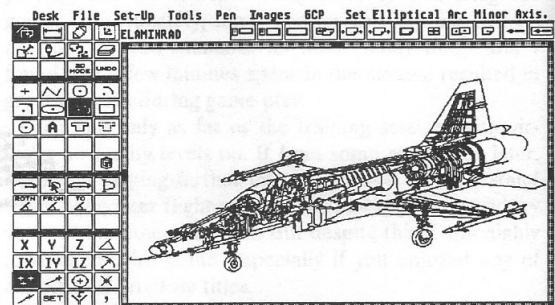
ScanArt from Migraph

A comprehensive clipart library, **ScanArt**, is available from Migraph. The two double sided disks in the package included about 100 images in IMG format. These images are compatible with Timeworks DTP, Supercharged Easy Draw, Word Up, and other programs that support this format. Most of the images are scanned at 150 dots per inch, meaning they are of very high quality and will look great when printed. **ScanArt** is \$49.95 from Migraph, 720 South 333rd (201), Federal Way, Washington 98003. (800) 223-3729. Below is a sample of **ScanArt**.



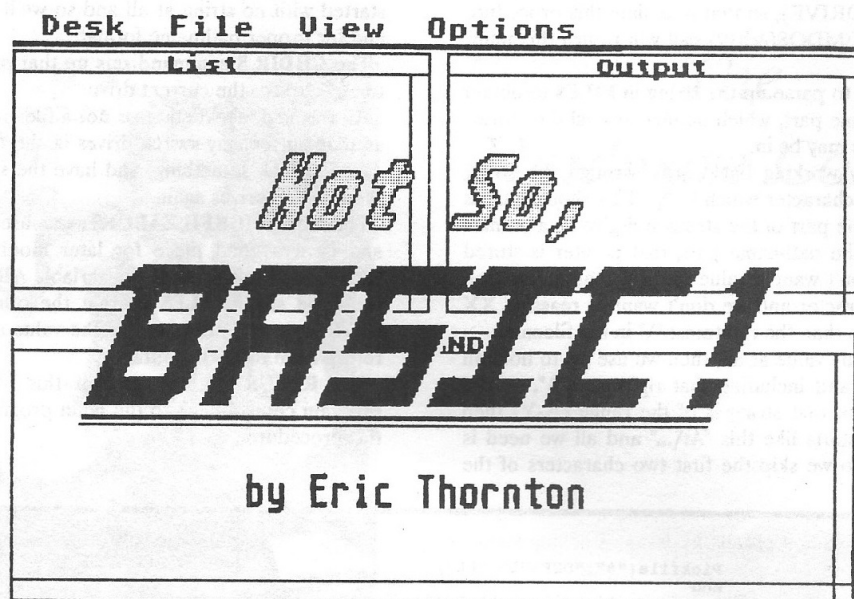
A Look at DynaCADD

DynaCADD is a brand new 2D and true 3D computer aided design and drafting (CADD) program from ISD Marketing in Canada. Supposedly, it is more powerful than either AutoCAD or VersaCAD, the two premier CAD packages for PCs. It features auto-dimensioning and DXF file format compatibility with AutoCAD. Here is a pictorial look at the package that sells for \$695 from ISD Marketing, 2651 John Street, Unit 3, Markham, Ontario, Canada L3R 2W5. (416) 479-1882.



DynaCADD in Action - Both 2D and 3D

END



Fileselector problem from GFA

The standard file selector routine in GFA Basic is fine, but it doesn't remember, the next time it's used, which drive and directory you switched to last time. There is a solution. Simply replace all your **FILESELECT** calls with calls to this procedure. It takes care of remembering which disk was last accessed, which directory on which disk, etc.

To call this procedure use this format:

Pickfile("ext","DEFAULT.FLE")

The first quoted string or string variable is the extender for the file search. Only files with this extender will be shown when the fileselector box is first shown. The second quoted string or string variable is the default filename automatically selected and displayed at the start.

In the procedure definition these strings are called **Ext\$** and **Def\$**. The variables **X**, **Xx**, **C**, and **A\$** are local, this means that those variables will not interfere with same name variables in your program.

The current drive (0,1,2,3,...) is returned by the GEMDOS call with the opcode \$19 (hex &h19). The current drive pathname is held in the built in array **Dir\$()**, it is indexed by a number which is (1,2,3,4,...), so we add one to the **CURDRIVE** number returned from GEMDOS and use that as the array index into **Dir\$()** to obtain **CURPATH\$**.

by Eric Thornton

If the first character of **CURPATH\$** is not in the range (A-Z) then we need to tack on a header which is of that set corresponding to the **CURDRIVE** variable. To turn (0,1,2,3,...) into (A,B,C,D,...) we take the ascii value of "A" and add to that **CURDRIVE**. That gets a character from the right set. A colon is added in between that character and the **CURPATH\$** returned above.

Finally the command we've all been expecting is used:

Fileselect Curpath\$ + "*" + Ext\$,Def\$,File\$

The current path is passed to **FILESELECT** along with "*" + **EXT\$** whatever the extender is, the default filename is supplied along with a string variable for returning the selected filename.

There are two conditions which would signal an abort status upon return to basic, one is **OK** when clicked, but no filename was selected, the other is **CANCEL** when clicked. In the later case, a blank string is returned **LEN(FILE\$)=0**. In the former case the last character of the returned filename is a "\" as in "A:\" or "C:\DATA\". If either of these is the case the variable called **ABORT** is set to one to indicate our status, so that later we do not try to open this filename. **ELSE** signals the code which is executed when the tested condition(s) are **NOT** true. In this case we clear the value of **ABORT** to zero.

MORE

The ascii value of the first character in FILE\$ is captured in C. The value of C is checked to see if it lies in the range A-Y and if so, the current drive is changed to this drive (CHDRIVE), so that next time this procedure is called the GEMDOS(&h19) call will return this number.

Now we need to parse up the string in FILE\$ to obtain just the pathname part, which defines any subdirectories the selected file may be in.

We'll start by working backwards through the string until we find a character which is "\". This should put us past the filename part of the string and give us a pointer to the end of the pathname part, that pointer is stored in XX. We don't want a value for XX that is less than the second character and we don't want to reassign XX to a lesser value than the rightmost "\" in the filename.

If XX holds any value at all, then we use A\$ to hold all characters left and including that rightmost "\". If the first character of that string is of the range (A-Y) then the pathname starts like this "A:\..." and all we need is the "\..." part so we skip the first two characters of the

string. Since we already know the current drive specifier, we tack that character and a ":" on the start. If the last character of the pathaname is ":" then we started with no string at all and so we'll put a "\" at the end for proper pathname format.

The CHDIR \$ command sets up that pathname (directories, etc) for the current drive.

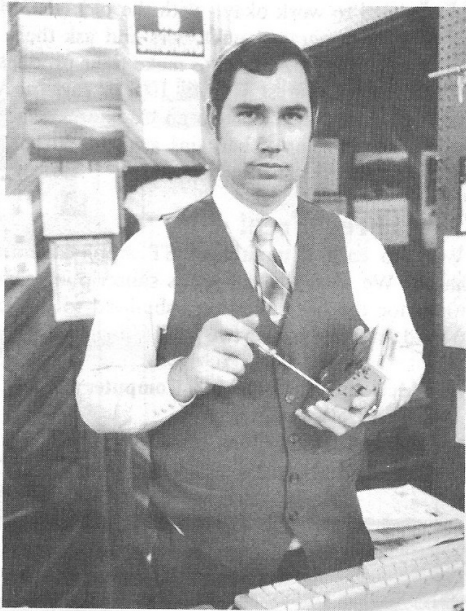
All this is done whether or not a filename was selected so that a user may switch drives in the fileselector, then cancel, check something, and have the system not make them switch drives again.

The label, PICKFILEABORT: was used during debug, and is at a good place for later modifications to this routine, so I left it in. If the variable ABORT is set then clear the string FILE\$ so that the other routines can check for a null string or for the value of ABORT upon return from this procedure.

The RETURN at the end of this procedure, sends program control back to the main program which called this procedure.

```
Pickfile("*.","DEFAULT.FILE")
End
Procedure Pickfile(Ext$,Def$)
Local X,XX,C,A$
Curdrive=Gemdos(&h19)
Curpath$=Dir$(Curdrive+1)
If Asc(Curpath$)<Asc("A") Or Asc(Curpath$)>Asc("Z")
Curpath$=Chr$(Curdrive+Asc("A"))+"."+Curpath$
Endif
Fileselect Curpath$+"*."+Ext$,Def$,File$
If Len(File$)=0 Or Asc(Right$(File$))=Asc("\")
Abort=1
Else
Abort=0
Endif
C=Asc(File$)
If C>Asc("@") And C<Asc("Z")
Chdrive C-Asc("@")
If Ret<>0
Else
Curdrive=Asc(File$)-Asc("A")
Endif
Endif
XX=0
For X=Len(File$) To 1 Step -1
C=Asc(Mid$(File$,X,1))
If C=Asc("\") And X>2 And XX=0
XX=X
Endif
Next X
If XX>0
A$=Left$(File$,XX-1)
C=Asc(A$)
If C>Asc("@") And C<Asc("Z")
A$=Right$(A$,Len(A$)-2)
Endif
A$=Chr$(Curdrive+Asc("A"))+"."+A$
If Asc(Right$(A$))=Asc(":")
A$=A$+"\\"
Endif
Chdir A$
If Ret<>0
Else
Curpath$=A$
Endif
Endif
Pickfileabort:
If Abort
File$=""
Endif
Return
```

(END)



The Hardware Column

by Bob McDonald

Joy Stick And Mouse Problems

Dear Bob:

I have both a joy stick and a mouse. When I plug them both in at once, certain programs refuse to run. What's going on?

Edward K. Ellington, Washington DC

The joy stick is usually the culprit. The ST's joy stick port is controlled through the keyboard, which sees some types of joy sticks as stuck keys. For example, the Turbo Quick Shot's rapid-fire button: if you leave it in the rapid-fire position when you boot the computer, you'll bomb. The same thing applies to the little keys that you use with Leader Board. The Pro-Line joy stick from Atari, which was designed for use with the 7800 video game, also can cause this kind of problem. Simply leaving certain models of joy sticks plugged into the computer when booting up can prevent access to certain programs. The best procedure when selecting a joy stick is to check it out on your dealer's ST with the software you plan to run.

Those Dropping Disk-Drive Doors

Dear Bob:

Every once in a while, when I pull out a disk, the drive acts as if I've inserted a disk. The door drops down, block-

ing the next disk's entry until I press the eject button. What's going on here?

Benjamin Carter, NYC

Don't just put up with this malfunction! If permitted to continue, it can destroy the alignment of your drive head and ultimately cause mechanical damage. Take the drive to your service center or contact Atari directly, to find out what repairs are needed. Usually it's just a matter of adjusting a few springs, after which the problem disappears forever.

Colorless Color Monitors

Dear Bob:

My new color monitor seems washed out, compared to monitors I've seen on my friends' ST's. What can be done?

James Johnson, Brunswick NJ

The ST's color monitors have been made by three companies: JVC, Gold Star, and Samsung. The Gold Star - which can be recognized by its single, overall gray color - sometimes shows inadequate color intensity. Any ST service center can easily adjust the monitor for proper colors. This is not a malfunction; Gold Star, for some unknown reason, decided not to ship all its monitors adjusted for full brightness.

MORE

Hard Drive Problems

Dear Bob:

I've had some strange problems since I installed my hard drive. I booted and then clicked on a floppy drive's icon. The window opened, announcing that there were no files on that floppy, which is incorrect.

James Lunceford, Fulton MO

These problems usually derive from having too many files in an open window in Drive "C" or "D," or whichever letter you've assigned to your hard drive. The operating system can recognize no more than 400 files at any one time in an open window. You have to keep track of how many files you have in any window you've opened, including those within folders.

Rom Incompatibilities

Dear Bob:

I just traded up to a 1040ST. Some programs - which I had no difficulty with before - just won't run. They boot up part way and then they hang. Why?

Kenneth Norvo, Beardstown IL

Some of the new 1040ST's are being shipped with the new "Mega" ROMs installed. There's a good possibility that you have them in your 1040 ST. Certain programs

won't run with these newer ROMs, which is almost always the software publisher's fault. Atari has warned against using undocumented commands, but some software authors used them regardless, simply because they seemed to work okay - with the older ROMs! Call the problem-programs' publishers and ask them if they can supply fixes or patches to make their products compatible with the latest ROMs. If they say, "no," try to find out when they expect to go Chapter 11 and let me know. There's a lot of bargains available at bankruptcy sales!

Free! Free! Free!

Want to earn 6 months of ST Applications free of charge? We award a half-year's subscription prize each month for the best question submitted to this column. So send in a question or two to:

Bob McDonald, McDonald's Computer Center
4921 East Colonial Drive
Orlando, FL 32803-4309.

END

Our Policy on Submitting Material

Contributions to **ST Applications** are welcome from everyone. We want a variety of articles and programs which can be helpful, useful or just fun for other Atari ST owners.

We prefer articles with accompanying programs (and hopefully, at least one graphic) which demonstrates a type of programming style or application. We wish to keep the content of the articles serious enough to enable readers to refer back to them in the future- something to build on. We feel this makes for a more meaningful relationship with the user and his/her computer. Chances are if a topic interests you, other people will find it useful as well. Our need for short articles is never ending.

All articles submitted to **ST Applications** become the property of **ST Applications** unless otherwise agreed upon by both the author and **ST Applications**. Should you not receive acceptance from us within two months please call or write. Should you desire remuneration please so state with your submission. Our normal payment for articles begins with either a 1-year subscription or \$35 (more, depending on topic, etc.). We pay \$15 for software, hardware and book reviews.

Please include both text and program files on your submission disk. Be sure to send a hardcopy of both the article and program listing or graphic (including photos and drawings).

Our address:

ST Applications

P.O. Box 980

Forestville, CA 95436

Our Phone Number: (707) 887-7879



Desktop Publishing:

Timeworks' Publisher ST

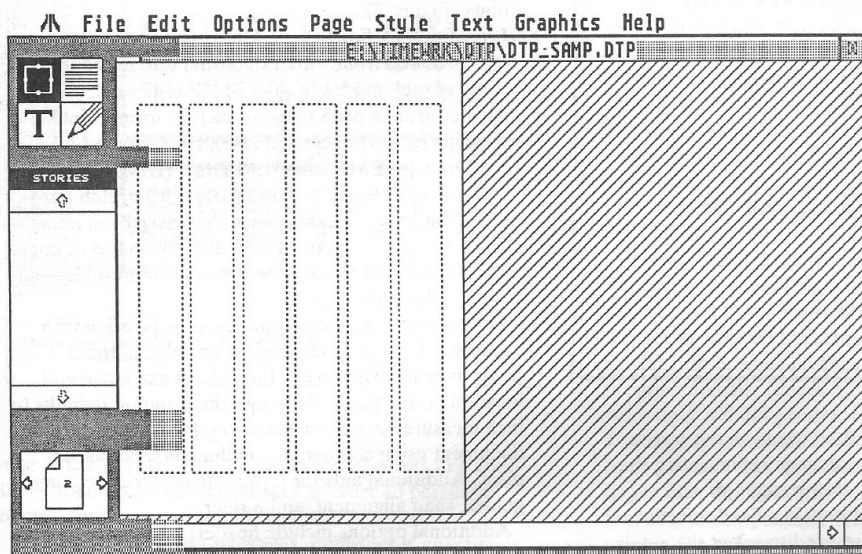


Figure #1

by Dan Fruchey

One of the most important new desktop publishing products released this year is *Timeworks' Publisher ST*. Originally intended for release last November, this clone of the PC program Publish It! boasts versatility and compatibility.

Timeworks has established a niche in the ST market by producing user friendly application programs that are logically arranged and competitively priced. Word Writer, Datamanager, Swiftcalc, and Partner have all been top selling products on the ST. **Publisher ST** promises to follow in their footsteps.

The ST version of Publisher was created for *Timeworks* by GST Holdings. This well known U.K. company has built their reputation on products such as 1st Word, 1st Word Plus, and the ST version of Word Writer. The efforts of both companies have been combined in Publisher to create a viable new product.

Out Of The Box

Publisher comes in a vinyl case identical to others available in the *Timeworks* collection. The manual is contained in a sturdy black binder and documentation of the program appears complete. The manual also includes tutorials on use and printed samples of the fonts and clip art enclosed with the program.

Publisher is supplied on six disks that contain a range of fonts and drivers suitable for every conceivable sys-

tem configuration. You can run the program on a basic 512k system with a single sided disk drive and a dot matrix printer or on any hardware combination up to a MEGA 4 with a hard drive and laser printer.

Before using Publisher, it's necessary to customize a copy that will take advantage of your system configuration. To configure Publisher, you simply format four disks and then run a utility called INSTALL.APP. The install program will ask you to select a system configuration (number and type of drives and printer type) and then tell you which disks to insert in order to create a customized version of Publisher.

Once this has been completed you must run a utility, FONTWID.APP, that checks the height and width of the fonts you are using. Installation takes approximately half an hour and it is quite painless.

How It Looks

Timeworks' Publisher ST uses the familiar drop down menus, mouse functions, and windows used by GEM. Publisher also allows keyboard access to most options using dual key macros. Where possible, commands parallel those used by other *Timeworks* products (i.e. ALT-B activates bold print in both Publisher and Word Writer, etc.). Most program options are fairly transparent and functions are clearly explained in the users manual.

MORE

Font Samples From **Publisher ST**

Swiss Font (Timeworks)

Dutch (Timeworks)

Rockface

Ravinia

DRURY LANE CAPS

MADISON

Figure #2

There are only two non-standard items on the screen. One is a box divided into four sections representing the four modes of operation: box, text, paragraph, and draw. The other non-standard item is a fixed selector box (Figure 1). These icons are used to select the various options used to create documents.

Up And Running

To create a document you must first define a page size (letter, legal, or half sheet) and orientation (landscape or portrait). A blank page in the chosen size will appear on the screen. Using a master page option and column guides, it is possible to define text locations on the initial and subsequent pages.

Once an area for text has been defined you can either begin typing by moving to text mode or directly import formatted documents from a wordprocessor. Publisher allows importation of text in ASCII, Word Writer, 1st Word, 1st Word Plus, or Wordperfect format. This is the first GDOS publishing program that allows direct importation of formatted text.

If an imported document is longer than a single column, it can be wrapped to other columns by indicating its destination. Using Word Writer, it is possible to "tag" text indicating whether it should be a portion of the body text, headline, or subhead.

Using the paragraph mode and style sheets it is possible to predefine text sizes, styles; and justification formats for headlines, subheads, body text, and bullets. Each time a future page is created, portions of text will default to your favorite options.

Once text has been placed on the page, you can adjust leading, word spacing, justification, kerning, and hyphenation. To speed up use, it is also possible to

define function key macros that will automatically execute selected commands.

Publisher includes a wider range of font families than any other program currently available. Swiss, Dutch, Ravinia, Rockface, Madison, Drury Lane, and Bullet fonts are all available in sizes varying from 5 to 72 points (Figure 2).

Line graphics, borders, and basic geometric shapes can be created from within Publisher and tinted in a variety of grey shades (Figure 3). Graphics can also be imported from a wide range of picture formats including image (.IMG), Degas (.PI1-3), Neochrome (.NEO), Easy Draw (.GEM), and GEM Draw (IBM .GEM).

It is possible to edit, cut and paste, and stretch portions of bit image graphics once they have been imported. Graphics can be placed directly on top of documents; text will automatically "runaround" graphics and continue down the page.

Extremely precise sizing and layout is possible with Publisher. By double clicking on any object, the program will advise you of the objects exact size and location on the page. Although the program defaults to inch measurements, you can also select to measure the document using centimeters, millimeters, points, or picas. Additional aids for layout include grids, column guides, snap alignment, and a ruler.

Additional options include headers and footers, creation of multi-page documents, multiple copies at print time, five zoom levels, page offset commands, and much more.

Personal Impressions

Timeworks' Publisher ST is sure to be a popular program. It is easy to use, versatile, and it can import text and graphics from almost every conceivable source. The program is priced competitively with similar pack-

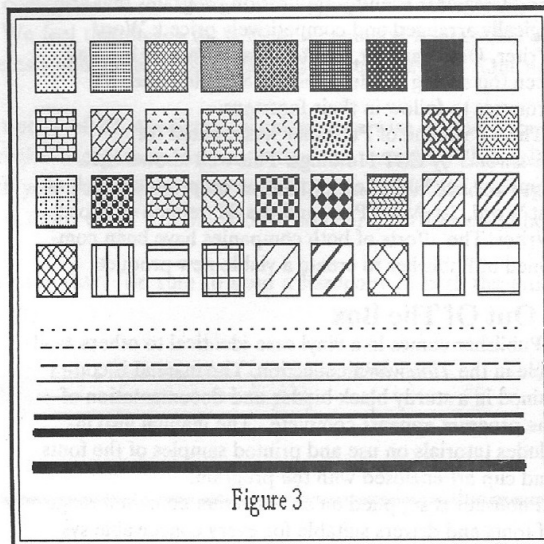
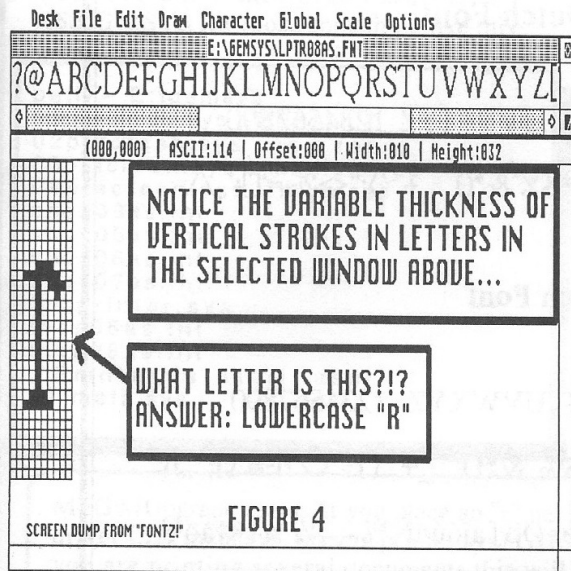


Figure 3



ages (\$129.95 retail) but the inclusion of extras like additional fonts and a clip-art library are sure to persuade many buyers to purchase Publisher.

Like most *Timeworks* packages, Publisher is geared towards the "average" user who has an occasional need for desktop publishing. Most documents, including newsletters, flyers, and brochures, can be easily created with this program.

I must confess that I purchased a copy of Publisher long before ST Applications ever received their copy. I am pleased with the level of text manipulation capabilities available. The ability to import formatted text has saved me endless hours of unnecessary reformatting.

On the surface, the graphic importation and manipulation capabilities of Publisher are acceptable. The only picture formats not accepted by the program are Tiny (.TNY, .TN1-3) and Degas Elite (.PC1-3); if you own pictures in either format they can be converted to another format Publisher will import.

Although I was impressed by these basic abilities I have also noticed the program lacks many options used in page layout and design. The program lacks alignment commands for organizing groups of objects in relation to each other and the page. Publisher does not support object grouping; if you need to rearrange a layout text and graphics must be moved and realigned separately (on complex layered graphics this could take hours). Object layering is primitive and difficult to perform. GEM graphics can't be edited or even separated from other GEM graphics on a page. Objects must be addressed individually for option changes. Each time an

object is created, it uses a series of default options that may be inconsistent with the options you preselected.

Many of the fonts supplied with the program are inaccurately sized or contain letters that have been improperly formed. This especially applies to the laser printer fonts. Characters should maintain a consistent weight (thickness). Publisher's fonts are inconsistent (Figure 4). I measured text and found 18 point fonts that were actually 19 points, 8 point fonts that actually measured 7 points, etc.

The fonts fail to maintain even height and width ratios among devices (monitors and printers). A document may appear correctly balanced on screen and print correctly on a dot matrix draft copy. When the exact same page is printed on a laser printer, the text may appear shoved to one side or contain irregular spacing that is difficult to correct.

Publisher allows some fonts to perform double duty. The ten point screen font is also used for the twelve point screen font. There is no way to visually tell the difference in sizes on the screen. There are no 300 dpi (laser) printer fonts on the disks. Period. The install program actually renames the 150 dpi fonts so they will work on a 300 dpi printer. For some uses, this is fine. The largest actual 150 dpi font, a 36 point size, converts to an 18 point font at 300 dpi. If you need a font that is larger than 18 points, the software enlarges an existing font size to compensate. With the resolution of a laser printer, the enlarged fonts appear extremely jagged and are essentially useless. To get larger sizes I was forced to add Atari's official fonts to my working disks. Even then, I noticed that some of the existing 150 dpi fonts I could use weren't even added to my customized disks.

The fonts only use symbols directly accessible from the keyboard. If you need a character from the extended character set such as a Greek or Hebrew letter, a copyright symbol, or an unlabeled "e" you're out of luck. These characters don't even exist in Publisher's font set. Atari's full font set defines 256 characters; Publisher's font set only defines 94 characters (see Figure 5 on next page).

In all fairness I should mention that there are "work around" methods for some of these drawbacks. However, some of these "work around" methods could take hours and an additional cash outlay for fonts and other programs.

If you only plan on using a single printer and you only need the keyboard characters, most of these font problems won't affect you. If you plan to send disk files to other ST users, take your documents elsewhere for printing (such as a dealership with a laser printer), or require special symbols, you had better carefully consider other options.

Bug Report

Like any new product, the first version of Publisher has a few bugs that need to be worked out. Most can be overcome if you are aware of them. A few bugs are com-

Timeworks Dutch Font

ACBDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

abcdefghijklmnopqrstuvwxyz !@#\$%^&*() _+~{}:"<>?|=~[];.,/\

Atari Dutch Font

ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

abcdefghijklmnopqrstuvwxyz !@#\$%^&*() _+~{}:"'<>?|=‘[];’,./\

ÇüëääåçêëèîïÄÅÉæÆôöùüÿÖÜøŒßíóúñÑ^{ao}ıŁı½¼ı«»ãõŸ

çœƎÀÃÕ˘†¶©®™ıȨגבאזחזהרנמלכטפסנמצקצףםךזתשרקצפסנמלכטחזהרגבאזח

$$\Gamma\pi\Sigma\sigma\mu\tau\tilde{\theta}\Omega\delta\phi\in\mathcal{N}\equiv\pm\geq\leq\int\div\approx^{\circ}\bullet\cdot\sqrt{n^2\,3}?$$

Figure #5

patibility problems caused by other programs and are not directly related to Publisher.

Perhaps the most important problem is incompatibility with the 520STFM. If you own Publisher and a 520STFM call Timeworks ASAP and request version 1.11 (they are currently shipping 1.10). If you own any other system, please don't call and request an upgrade; there are no other changes in the revision.

The current version has an option for printing multiple copies; if you indicate that you wish more than one copy to be printed, the additional copies come out garbled and incomplete. Timeworks is aware of the bug and plans to fix it in the next major upgrade.

If you are trying to run more than one GDOS program using Publisher's ASSIGN.SYS file, you may find that other GDOS programs bomb or won't load. Why? When INSTALL.APP writes the ASSIGN.SYS file it inadvertently fails to write a line that reads "00p SCREEN.SYS." You can fix this with a text editor (Figure 6 shows where the line of text should go).

If you have tried importing Easy Draw .GEM files into Publisher you may find that they print without some line and fill styles or appear distorted on screen. Migraph used some non- standard line and fill patterns in creating Easy Draw. If any of these fill patterns (the last three on Easy Draw's pattern list) are in a GEM

graphic it will load improperly. Also, GDOS text will not be displayed within an imported graphic. To load individual .GEM graphics (like clip art), remember that pictures must be saved in separate files.

Although this does not affect program use, the manual index is poorly laid out and many items are missing. In several cases, it required a great deal of page flipping before I could find information on specific commands and functions.

Talking To Timeworks

I recently spent almost an hour talking to a Timeworks support tech, Paul Sevigny, on the phone. He was friendly and extremely helpful. Unfortunately, his answers were less than satisfying in many cases.

I asked about the number of disk swaps required on a single drive system. When I first tested Publisher I was using a double sided disk drive and a 2.5 Mb ST (upgraded 520). I asked why Publisher wouldn't customize the program to use all of my RAM. Paul said this was intentional. To guarantee operation on all ST and MEGA systems the program is based on the lowest common system denominator: the 520 ST. A 520 requires the large number of disk swaps because of RAM limitations. Currently there are no plans to allow fewer disk swaps and effective RAM use on larger systems.

Your current Timeworks Publisher ST ASSIGN.SYS file looks something like this:

```
path = E:\gemsys
01p screen.sys ; DEFAULT
02p screen.sys ; low res
03p screen.sys ; medium res
04p screen.sys ; high res
shss03as.fnt
shss05as.fnt
shss06as.fnt
shss07as.fnt
21 printer.sys
lpss06as.fnt
lpss08as.fnt
; Printer is Atari laser
31 meta.sys
```

Add the line "00p screen.sys" in the space indicated below:

```
path = [your disk drive]:\gemsys
00p screen.sys
01p screen.sys ; DEFAULT
02p screen.sys ; low res
03p screen.sys ; medium res
[I.D.'s of medium res screen fonts]
04p screen.sys ; high res
[I.D.'s of high res screen font]
21 printer.sys
; Printer is [your printer]
[I.D.'s of printer fonts]
31 meta.sys
```

MEGA/Upgrade users: If you place an "r" behind the I.D. number of the printer driver or metafile driver (31r meta.sys, or, 21r printer.sys) GDOS will load and retain the drivers at boot time. If you are printing several documents this will save time normally used to reread drivers before printing. This procedure is not recommended for 512k and 1Mb machines. Publisher ST may need this extra memory for pictures, etc. on these machines when a document is being created.

Figure 6

I have heard reports that 1040 owners are allowed a wider range of fonts than 520 owners. Apparently this is true. The 520 doesn't have as wide a range of font sizes due to RAM limitations (seems logical). The kicker is that an upgraded 520 will be configured the same way as a standard 520 is with fewer fonts. Why? Paul said that when the install program reads system information from TOS, all 520 systems, regardless of memory capabilities, appear identical. A 1040 or MEGA allocates TOS differently and the install program allows more fonts to be used. It is possible to add or delete fonts, if you know GDOS, but otherwise you're out of luck (next month we'll discuss GDOS and I'll explain how you can add those additional fonts).

One of my last questions related to users of Postscript and other printers that are accessed via a serial port (such as the original HP Laser Jet). I had been told that files couldn't be sent to printers that use a serial port. Paul tried to explain a complicated method of how I could use a couple of parallel to serial cables and a modem to make it work. I interrupted and asked if they would make the program serial compatible in a future revision. The answer? No. Not in the foreseeable future.

Except for bug fixes *Timeworks' Publisher ST* will remain essentially unchanged for an unknown amount of time. When the IBM and Amiga versions are upgraded, the ST version will also be given a face lift. My biggest question is whether they consider the font problems a bug or an upgrade problem.

Final Synopsis

Publisher ST is easy to use and priced reasonably. Extras such as a clip art collection and multiple fonts make it an attractive product. The program uses GDOS fonts and drivers. It is currently the only GDOS program that allows importation of wordprocessed documents. Publisher allows importation of graphics using most formats and also allows you to create and edit graphics from within the program.

Drawbacks to program use include some inaccurate fonts, a reduced font set (only 94 characters), and restricted page layout and object manipulation capabilities. The program does have a few bugs but they are being fixed as rapidly as possible.

To the best of my knowledge, **Publisher ST** is the only publishing program that runs on a 520 ST or 520 STFM. I highly recommend it to new desktop publishers, occasional users of desktop publishing software, and those who plan on restricting use to standard documents such as newsletters and flyers. **Publisher ST** is easy to learn and, compared to other publishing programs, easy to use.

Publisher ST retails for \$129.95. Publisher is also available directly from *Timeworks*. To order call: (312) 948-9202. Or write them at: *Timeworks*, 444 Lake Cook Road, Deerfield, Illinois 60015.

END

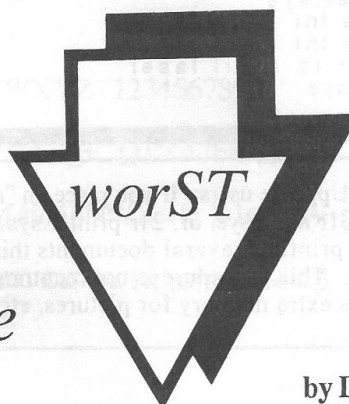


The



beST

and the



worST

by Dick Biow

WORD PERFECT UPDATE

Some months ago an industry newsletter announced that the WordPerfect Corporation intended to drop its Amiga version because of a "possible saturation of the market." Don't believe it. Don't worry about the Atari ST version's lifespan either. Both versions will be around a long time.

According to Dan Lunt, WP Corp's Vice President for Sales, "I don't think you'll ever find us removing a product from the market because we think we've sold as many copies of it as we are likely to sell. We are concerned with making our Amiga and Atari divisions more profitable, but that simply means we have to figure out how to develop this product at less cost to ourselves.

"We're already starting to accomplish this . . ."

INTERRUPTION!

Before we let Dan continue, let's explain something you most certainly know, but your cousin, who has just pulled this magazine off of a rack, may not know. The Macintosh, the Amiga, and the Atari ST are all built around Motorola's 68000 CPU chip. This means that three different people writing in assembly language for these three different computers will all use Motorola's set of commands for telling their CPU's what to do. If these programmers are all writing the same program--say, WordPerfect--for these three computers, you will be able to find places where their assembly code printouts are going to look quite similar to one another. If you are a member of management, this will lead you to wonder whether you are paying people to duplicate one another's work. This will lead you to ask yourself . . .

"... why can't those 3 three operations be streamlined?" We now return the floor to Dan Lunt:

"We cut our programming staffs for the Amiga and Atari divisions, moved the freed programmers into our Macintosh division, where they're working on modularizing the software being developed by the Macintosh division. That means that the WordPerfect code for the Macintosh will be easy to port to the Atari and Amiga. Then the programmers that are still assigned to the Atari and Amiga will do whatever fine-tuning will be needed for the user interfaces and system calls, to make WordPerfect operate correctly in the Atari and Amiga environments. This should cut down our development costs considerably and help us work more profitably in those areas.

"The ST and Amiga markets are similar: sales are fairly low. After talking with Atari and Amiga, we're finding that they are not selling the numbers of machines that would sustain an enormous number of sales of WordPerfect. We find most of our sales are to folks who just bought their computers. They aren't going back later and buying a word processor; they usually buy it at the same time they buy their machine.

DOES WORDPERFECT HELP SELL ATARIS?

"I expect that the availability of WordPerfect would influence people who are considering the purchase of Ataris or Amigas. I would think that the availability of WordPerfect for those machines would attract some of the people who are looking for high-end word processing and are perhaps a little bit undecided as to which machine they would want to buy.

"At the same time, you have to realize that the Atari and Amiga markets are different from the PC market. We see different applications for the Atari, with a lot of emphasis on music. The Amiga is being used a great deal for video reproduction. This tends to put those machines more into specialty markets. The number of people who are buying those machines in order to put them into business environments is much smaller than what we find for the PC market."

DESKJET UPDATE

All the nice things we said about the Hewlett Packard DeskJet in the June issue are true: as proof,

consider that this column -- the one you are looking at now -- was printed out on a DeskJet with WordPerfect. As we said, you can get it for \$700 or less, it prints at 120 characters per second, it has the same 300-dots-per-inch as the HP LaserJet, and you can't tell its printout from a laser printer printout... uh, well...

You can't tell its **fixed-width-character** printout from that of a LaserJet. In fact, HP's publicity kit gives you samples of printouts from both printers and challenges you to tell them apart, but they are careful to use **fixed-width** Courier type. What they don't tell you is what Susan Robison discovered, which is that there darn well **is** a visible difference if you use a proportional type style and ask for an even right hand margin (right hand justification). Because the DeskJet can't do "kerning."

Susan Robison writes printer drivers for the WordPerfect Corporation. When she was writing the DeskJet drivers, she noticed that -- unlike the LaserJet -- the DeskJet can't "kern," which is to say it can't do things like shoving the letter "a" part way under the right arm of a capital "T," which is a kind of adjustment that can make the right-justification of proportional type more elegant. Understand, this inability doesn't make proportional-spaced type on a DeskJet inadequate; it just doesn't look quite as smooth as what a laser printer can give you, and italics type won't justify correctly. You'd never notice the difference, unless you compared printouts of proportional type. And now you know why Hewlett Packard's publicity kit only lets you compare samples of **fixed-pitch** type from the LaserJet and the DeskJet!

Thanks, Susan!

PIRACY UPDATE -- LESSONS IN CHEATING

Last month we listed 7 myths about piracy. An eighth myth would be, "All we need to do is **educate** people, so that they realize that pirating software is as dishonest and contemptible as shoplifting."

Educate people? Ha! A software publisher informs me that the worst piracy he experiences is from elementary and high schools, which is to say basic **educational** centers. Many teachers take the position that they can't afford copies of a program for all of their students, so -- why shouldn't they do what

so many marching-band directors do on Xerox machines: make copies by the dozen?

So what we have is a generation of students who are learning from America's teachers -- our standard-setters, our role-models -- that piracy is perfectly ethical if you're convinced your cause is just (which is the old end-justifies-the-means argument in academic robes).

Once those kids graduate, how are you going to **educate** them to respect software copyrights? Raise money for a PR campaign by software publishers? Pay for a campaign large enough to compensate for the example set for students all through their school years? I repeat, ha!

Yes, sir or madam, I know. You are a teacher, and you **never** pirate software. You are to be commended. But all it takes is just one bent teacher in a student's entire career to convince him that whether or not to practice piracy is purely a matter of individual conscience, not social responsibility.

The conclusion remains as presented before, then: no practical solution for piracy has appeared, and none is likely to appear. What about those ST software publishers who are forever threatening to quit unless piracy disappears? We should encourage them to do so, and the sooner the better. Take your programs off the market, guys, and take your complaints off Compuwhine, and leave the field to the winners among us, who have the courage to recognize that piracy is simply part of the cost of doing business in the computer world, just as shoplifting is for department stores.

HOODWINK ALERT

You open your mail one morning to read this:

EXPRESS MESSAGE! YOU HAVE BEEN CHOSEN AS A RECIPIENT IN A NATIONAL GIVEAWAY/STOP/ YOU ARE CHOSEN TO RECEIVE ATARI DMP 2000, WITH DISPLAY, MEMORY, KEYBOARD, AND PRINTER OR \$1000 IN CASH/STOP...

So you /STOP/ and start searching for the fine print. Which says you must meet eligibility requirements listed on the reverse side of the page and call an 800 number. You flip the page. Ahah! To be eligible, you and your spouse must commit yourselves to accepting a 90-minute guided tour by a

representative of the Sunny Heaven timeshare resort, who will try to sell you a one-week-per-year ownership of a vacation cabin by the shores of the Oldman River. Whether you get the \$1000 or the Atari DMP 2000 will be up to the Sunny Heaven honchos, so you know you won't get the \$1000. After all, you're not **that** dim.

Still, it might be worth 90 minutes of your time, plus transportation, to "RECEIVE ATARI DMP 2000 WITH DISPLAY, MEMORY, KEYBOARD, AND PRINTER," mightn't it? After all, the "express message" also refers to the Atari DMP 2000 as a "DESKTOP ATARI." Isn't a desktop Atari worth 90 minutes of listening to a salesman's spiel, plus maybe an 90 minutes drive each way?

No. Not unless your time isn't worth much.

WHAT'S AN ATARI DMP-2000?

In the first place, even though an Atari DMP-2000 is referred to as a "desktop Atari" in the letter, it is definitely not a desktop computer. In fact, it isn't even a computer, nor is it a desktop anything. The Atari DMP-2000 is a **hand-holdable printing calculator!** It's a nice, reliable unit, well-made and accurate, and worth every penny of its \$24.95 list price. If obtained free of charge, it would make a great little gift -- only you aren't going to get it free of charge.

The finest fine print at the end of the letter states, "Recipients must pay all taxes on gifts. These gifts will be sent to the recipient, who will be charged a reasonable fee for shipping and handling."

What's a "reasonable fee?" Well, if "desktop Atari . . . with display, memory, keyboard and printer," is a reasonable description of a DMP-2000, do you really want to find out what a reasonable "fee" is likely to be? You don't? Then, when you get one of these letters, you'll know where to file it, won't you?

DECELERATING THE ACCELERATOR BOARD

When Megabyte Computers' Accelerator Board was first announced, there was a lot of excitement among the guys around our local Atari shop. By about the second week in March, according to our local bulletin board, we'd be able to buy upgrade

boards that would run our Ataris at 16 megahertz, as opposed to the present 8 mhz. The price would be about \$200-\$250, and "some soldering" would be required.

I immediately called Megabyte, where Kevin Henderson provided a candid overview of the Accelerator Board. Installation would not merely require "some soldering." It would require "substantial soldering." Nor would installation be a job for a novice; only selected Atari repair centers would be authorized to do the job. As for the delivery date-- well -- new uncertainties had appeared. The present mockup worked fine in the U.S., but tests had shown that certain European programs which expected 50-Hz European current would confuse it.

"We know how to correct the problem, and we will," said Kevin, "but, you understand, we can't promise deliveries until our board is in production." He had a point, which is that, once a piece of hardware is being manufactured, you can't correct it cheaply with upgrades the way you can do it with software. You have to get everything absolutely right before you begin fabrication. If you don't, you may have to do what IBM did with their first keyboards for the PC Junior -- "eat them," as they say in the trade. This is economically feasible if you happen to be IBM. If you happen to be Megabyte Computers, it's disastrous.

Furthermore, said Kevin, the price for the Accelerator would be \$50 more than the original gossip had suggested. And, even with the higher price, and even with the production delay, there was a long order list building, interviews in major ST magazines in the works, and high hopes all around. I asked to be put on that order list, hopefully as high as possible, as I intended to review the board.

A week ago I got climbed off the list.

YOU TRY IT FIRST!

I look at it this way; I know the board will work, and I know Megabyte Computers means business. But I know some other things and infer from them that I don't want to be the first guy in town to have an Accelerator Board installed in my ST.

Megabyte has taken so long to get the board to market that problems must be greater than had been predicted. This makes me uncertain that the first run

USE PC COMPATIBLE HARD DISKS

WITH THE ATARI ST COMPUTER!

THE BMS-100 HARD DISK SYSTEM ALLOWS YOU TO CONNECT READILY AVAILABLE PC COMPATIBLE HARD DISKS THROUGH THE DMA PORT OF YOUR ATARI ST COMPUTER

WITH THE BMS-100 SYSTEM, YOU CAN

- READILY CONFIGURE YOUR HARD DISK SUIT YOUR INDIVIDUAL APPLICATION
- HAVE THE OPTION TO EASILY UPGRADE FOR GREATER STORAGE/PERFORMANCE
- CONTINUE TO MATCH YOUR SYSTEMS CAPABILITIES WITH YOUR CURRENT REQUIREMENTS FOR A MINIMUM OF COST
- UTILIZE THE BUILT-IN CLOCK/CALENDAR TO HELP ORGANIZE YOUR FILES

For orders/information, call

FOR ONLY:
\$250



**BERKELEY
MICROSYSTEMS**
360 Oakland Ave, Suite 5
Oakland, California 94611
TEL (415) 465-6956

SOFTWARES WEST Introduces The HOLLYWOOD Font

A commercial quality GDOS font for users of Easy Draw, Timeworks Publisher, Fleet Street Publisher, Word Up, Microsoft Write, Degas Elite, and other products that use GDOS fonts. The Hollywood font uses the ST extended character set (179 characters defined). Includes 7, 10, 12, 18, and 24 point font sizes. Available for 9 pin (Epson), 24 pin (180 dpi), HP (150 dpi), and Laser (300 dpi) printers. Please specify your printer type when ordering.



\$19.95

For Single Sided disks add \$3.00



Send Your Check or Money Order To:

SOFTWARES WEST
P.O. Box 2846 - A
Laguna Hills, CA 92654-2846

California Residents please include 6% sales tax

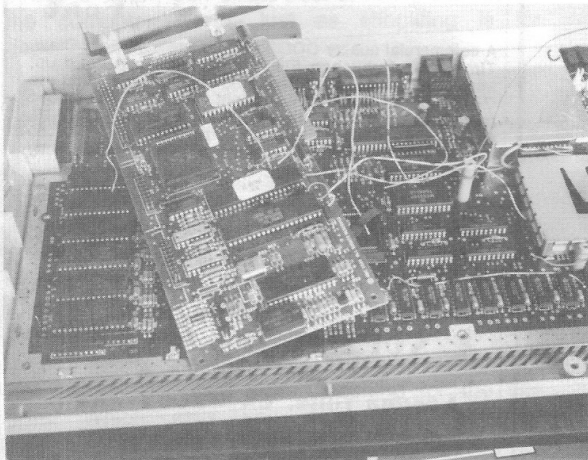
We also produce a variety of clipart collections in Degas (.PI3), Image (.IMG), and GEM (.GEM) formats. Write for more info.

Dealers Inquiries Welcome

of production boards will precisely match the board that Megabyte breadboarded and demonstrated. No, I'm not afraid that Megabyte won't stand behind its product if something goes -- or already is -- wrong with a board. I'm afraid that I'll never know whether my problem lives in their board or in my motherboard!

Look: right now, when my 520ST gets into a snit and won't run properly, I take it into an authorized Atari repair shop. The repairman removes the Tech-Specialties memory upgrade, plugs in Atari's diagnostic cartridge, and checks the machine's operation. If the problem remains, he homes in on and then reseats those particular chips, like the MMU, that forever seem to want to sneak out of their sockets. On the other hand, if a performance glitch appears only when the memory upgrade board is plugged in, he knows the problem lies in one of its memory chips or its contacts with the main board. From there on it's an easy walk.

Now suppose that, after an Accelerator Board is soldered in place, a similar performance difficulty develops. Atari's diagnostic cartridge doesn't know how to diagnose the Accelerator, so the repairman won't have an easy time hunting down the problem's source. Furthermore, he can't just unplug the Accelerator like a memory upgrade, in order to isolate the problem, either.



Yeah, sure -- he can unsolder a dozen or so connections between the Accelerator and the main board, segregate the trouble, cure it, and solder everything back together again. I don't know what your repairman charges per hour but, if he has to keep a soldering iron hot, you are likely to end up with a bill that might make you think you've been retaining a dentist.

This complication makes me want to sit back until a couple of hundred Accelerators have been installed in other ST owners' boards. By that time, experience will have made diagnosis easier and faster for Atari technicians. I want to be a computer operator, not a pioneer.

Then, on the horizon, there's the blitter chip. One day, Atari is going to sell blitters, probably via complete main board swaps. But Atari won't swap for boards that have been modified and, even if a previously installed Accelerator has been removed from a main board, they won't enthuse over receiving a board with empty solder connections where solder connections don't belong. (I know I wouldn't accept such a board from **them!**) So the ideal time to install an Accelerator in a 520ST or 1040ST would seem to be **after** a blitter-chip-equipped board has been installed.

There it is. There's good reason to believe that the Accelerator Board will be a great success, because it fills a real need. But not for me. Not yet. Sure we all want speed, but we don't need suspense, right?

SUPRA FD-10 UPDATE

Because you read this column every month, you know that Supra's FD-10 is a 10-megabyte "hard" drive that actually accepts interchangeable 5.25-inch floppies. It runs at about half the speed of Supra's **hard** hard drive, which is comfortably fast -- about the same as a PC-clone's hard drive. The one we've got runs comfortably enough to be taken for granted, and the halving of the true hard-drive speed rating causes no discomfort, no annoyance, no impatience.

But now Mark White of the Supra Corporation tells me he thinks that Atari's newest ROMs are going to speed hard-drive access enormously. So my FD-10 will run about as fast as your conventional Supra drive presently runs. And your conventional Supra drive will speed up to a point where you'll have trouble telling it from a RAM disk!

(END)



The Joys of Modula-2

Memory Location	Name	Meaning
\$42E	phystop	End of memory
\$436	memtop	End of user's program memory
\$446	bootdev	Which drive was used to boot
\$44C	sshiftmod	Screen Resolution
\$45A	colorptr	Pointer to color palette
\$45E	screenpt	Start of video RAM
\$4AC	saverow	Cursor line value
\$4BA	hz200	200 Hz system counter
\$4CA	autopath	Autoexecutive path

Figure 1 Favorite Low Memory Locations

By Sol Guber

It is seldom that I write a column at the request of the editor, but this column was written specifically for Bill Petry, to sell more magazines. The program is a very simple one that shows you interesting graphic figures on the screen. If you watch very carefully, there is a message flashed on the screen that says "BUY ST Applications." As always, there are several pieces to the program that make the method used an interesting problem. First the program contains three screens and explains how to shift from one to the other. Also, it uses the 200 Hz timer that is built into the system to determine how often the message is to be flashed and how frequently the other two graphic figures should be switched.

One of the most ingenious parts of the ST system is the ease of screen flipping. This ability was used in the very first drawing program for the ST. As a bonus to all of the people that purchased the first Atari's, there was a drawing program called NeoChrome that came with it. The program was moderately useful and evolved into the present one that can be purchased. Think back to those early days, when it was the first and only drawing program available. The most amazing part of the program was the UNDO key. You could make a drawing operation, and if you make an error, you could press the UNDO key and the error disappeared. Even a complicated operation like a fill that escaped its boundaries

could be undone. This UNDO was a surprisingly easy operation and was duplicated on most of the future drawing programs.

The reason it was so easy is that there is no part of memory that is defined as only to be used as screen memory. The screen does need 32000 bytes of memory for storage of the information, but it can be anywhere in the memory. There are a series of XBIOS commands that can be used to determine the location of screen memory and to set its location. There is also another function called the screen's logical base. It is here, at this location, that all of the drawing operations can be done. Also, there is no reason that you cannot show a screen and draw something in another portion of memory.

So how was this fast, miraculous UNDO performed? Very simply. There were two areas set up as screen memory, let me call them `s_work` and `s_backup`. If a drawing command was performed, it was done in `s_work` and if another drawing command was performed, then the first one was repeated in `s_backup` and the second was performed in `s_work`. If the UNDO key was pressed, then the system would flip to `s_backup` and the pointers would be reversed to `s_work` and `s_backup`. A very clever and easy system to implement. The program uses an array to contain the three address

of the various screens and also flips from one to the other under control of a clock.

The clock that is used is the 200 Hz timer, whose value is found at 4BA in memory. As the system clock crystal vibrates, 200 times per second, the number in that memory location changes. To use this timer, all that needs to be done is read the value in that location. There is the ability in Modula to specify the location that you wish a variable to reside at. It is done as follows in the program:

```
hz200 [4baH] : LONGCARD;
```

So now all that needs to be done is set another **LONGCARD** to the value in hz200 and comparisons and timing events can easily be done.

Unfortunately, it is not quite as easy as that. There are a number of reserved locations in memory that cannot be accessed by the ordinary programmer. Motorola, when they programmed the 68000 chip, has decided that much damage can be done by allowing some memory locations to be accessed easily, so that when you try to simply read the value in this location, you will generate an error condition. However, there is a method to access this location. There is a condition called the supervisory mode, where you can do things that cannot be done in normal circumstances.

The 68000 chip has a supervisory mode with its own stack and registers that is generally used for the system programming. It even has its own additional set of instructions. The **Supervisor** in the 68000 chips allows for multi-tasking, in a much more logical, simple method. There are two methods to access the **Supervisor**. One through **GEMDOS** and one through the **BIOS**. With the **GEMDOS** method, you can set the system in the supervisory mode, perform the actions that are desired, and then come out of the supervisory mode. In the **XBIOS** method that we will be using, what will be done is telling the supervisor to run a procedure, and then return.

In order to have a procedure that can be run through the supervisor, several conditions must be met. The first condition is that there is a return at the end of the procedure. If you have been writing any procedures in Modula-2, this last statement may not make sense. All procedures have **RETURN**s written at their end, so that the system knows when to come back to the calling procedure, and then to continue. Yes, but when a procedure is to be run by the supervisor, it is different. For procedures to be run in the supervisory mode, this must be done explicitly, since the normal compiler options need to be modified so that the **RETURN** is not automatically written.

When most programs are compiled in Modula-2, the standard options are used, and it is very seldom that any modification need to be made. However, there are four compilation options that can affect the object code that is processed. These options are known as 'C', 'P', 'S',

and 'T'. The 'C' option makes sure that all variables are set to zero when the program states. The 'P' option is used to generate the normal entry and exit code for procedures. The 'S' generates the code at the entry of each procedure to perform stack checking to make sure that there is no overflow. 'T' is used to generate the code to perform range checking on array subscripts.

These options are turned on and off by use of comment statements, with the '-' being used to turn off the option, the '+' used to turn it on, and the '=' being used to restore it to the former state. The '\$' is needed to show that these are compiler options. Several examples are:

```
(*T- turn off range checking *)
```

```
(*T+ turn it back on *)
```

```
(*T= turn it back off *)
```

```
(*S-, $T- turn off stack and range checking *)
```

The procedure that is used to get the value in the 200 hertz timer is called **GetTime**. Before the procedure is compiled, both the stack and the entry and exit code options are deactivated. For this reason, the **CODE** statement (4e75H) is used to make sure the code has a proper exit. Of course, these options must be turned on after the procedure is compiled. This is a very important part of the procedure.

The rest of the program is fairly easy to explain. I used several patterns for drawing from an article called **Modified Shapes for IBM** by Paul Carlson from the May 1986 **COMPUTE!** magazine. They are simple graphics making programs that use the **MOD** function. The only real innovation in the program is during the declaration of the variables. For the first time, I have used the [] notation in a declaration as follows:

```
hz200 [4BAH] : LONGCARD;
```

This means that hz200 is a long card and the place in memory that it is to be stored is location 4BA. Since this location is already used for the counter of the 200 Hz signal, what we have essentially done is produce a statement corresponding to a **PEEK** in BASIC. Look at a specified memory location and determine the value there. This ability to specify a memory location was an addition that Wirth made in going from PASCAL to Modula-2. Since the location is in low memory, the only way that a program can fetch this value is through the **Supervisor**.

Let me skip down to the initialization procedures, before I explain the loop that performs the actions. After the handle is determined, three different screens are set up. The memory location of the actual screen is determined and the other two screen are set up by subtracting \$8000 (32000) from the value returned from the **XBIOS** function. A better method would be to ask the system to allocate the proper amount of memory for these two screen and then to be sure that the beginning

of the screens are on a 256 boundary, but it was done in this method for simplicity. On a 520 ST with many accessories, this might cause a serious problem, since there is no checking to be sure that the program or other information is not being wiped out.

To write the message, the screen is set to the first location, the size of the letters is increased, a color is chosen, and the lines are written. The size of the letters is set back to normal. The screen is set to the second location, and the procedure **Curly** makes the pattern on the screen. The screen is set to the third location and the **Swirl** pattern is now placed there. Now the **Subliminal** procedure is called.

Subliminal is an endless loop that can be exited by pressing any key. The first operation is to use

SuperExec function to determine the time. Two variables are set to this value. The keyboard is then checked. If a key is pressed, then return. If the difference between time and switch is greater than 2000 (ten seconds have elapsed) then switch the screens between the two patterns. Set up the proper screen and monitor the time, until time is greater than time1. Increase time1 by the duration that we wish our message to flash. Change the screen to the message and then monitor the time again. Switch back to the design screen for the rest of the second and keep repeating until the message gets through.

Next month will be another drawing program to put curves on the screen using carefully defined mathematical formats. Now if you see graphical screens of pretty designs, don't forget to **BUY St Applications**.

```

MODULE Part25;
FROM AESApplications IMPORT AppInitialise,AppExit;
FROM VDIOutputs IMPORT GraphicText,PolyLine;
FROM VDIAttributes IMPORT SetGraphicTextColour,SetColour,SetAbsCharHeight,
SetLinColour;
FROM AESGraphics IMPORT GrafHandle;
FROM VDIControls IMPORT ClientWorkstation;
FROM VDIAttributes IMPORT SetScreenBase,ScreenLogicalBase,SetScreenBase,
SuperExec;
FROM GENVDIBase IMPORT PxyArrayType;
FROM SYSTEM IMPORT ADDRESS,CODE;

CONST
  lineal=" BUW";
  line2="ST Applications";
  dur= 3;

VAR handle,i,sc:INTEGER;
ch:CHAR;
time,time1,switch:LONGCARD;
hz200 (4000); LONGCARD;
screen ARRAY[0..2] OF ADDRESS;
Pxy: PxyArrayType;

PROCEDURE Curly;
VAR k,i,j,n,m,ii,jj,c,p,nj:INTEGER;
BEGIN
  FOR i:=0 TO 3 DO
    FOR j:=0 TO 3 DO
      IF (i MOD 2) = (j MOD 2) THEN
        Y[1]:=50; Y[2]:=0; Y[3]:=50; Y[4]:=0;
      ELSE
        Y[1]:=50; Y[2]:=0; Y[3]:=50; Y[4]:=0;
      END;
      X[1]:=40; X[2]:=40; X[3]:=160; X[4]:=160;
      FOR i:=0 TO 22 DO
        Pxy[0]:=X[4]+i*140;
        Pxy[1]:=Y[4]+j*50;
        FOR m:=1 TO 4 DO
          Pxy[2]:=X[m]+i*140;
          Pxy[3]:=Y[m]+j*50;
          k:=SetLineColour(handle,m MOD 2)+1;
          PolyLine(handle,2,Pxy);
          Pxy:=Pxy[3];
          Pxy:=Pxy[4];
          Xd[m]:=X[m] DIV 8+((X[m]*7) DIV 8);
          Yd[m]:=Y[m] DIV 8+((Y[m]*7) DIV 8);
        END;
        FOR p:=1 TO 4 DO
          X[p]:=Xd[p]; Y[p]:=Yd[p];
        END;
      END;
    END;
  END;
END Curly;

PROCEDURE Swirl;
VAR k,i,j,n,m,ii,jj,c,p,nj:INTEGER;
X,Y,Xd,Yd:ARRAY[0..4] OF INTEGER;
BEGIN
  ii:=1; c:=1;
  FOR j:=0 TO 3 DO
    FOR i:=0 TO 6 DO
      jj:=jj+1;
      IF (jj) OR (i>(6-j)) THEN
        ELSE
          IF (j<2) OR (i>2) THEN c:=(c MOD 3) +1; END;
          IF j=3 THEN c:=(c MOD 3) +1; END;
          X[1]:=0; X[2]:=80; X[3]:=160;
          Y[1]:=0; Y[3]:=0;
          IF i=3 THEN Y[2]:=48;
          ELSE Y[2]:=40;
        END;
      END;
    END;
  END;
END Swirl;

PROCEDURE Subliminal;
BEGIN
  handle:=GrafHandle(i,i,i,i);
  ClearWorkstation(handle); screen[0]:=ScreenPhysicalBase();
  screen[1]:=screen[0];
  screen[2]:=screen[0]-8000H;
  SetScreenBase(screen[0],screen[0],1);
  SetAbsCharHeight(handle,24,i,i,i);
  ii:=SetGraphicTextColour(handle,2);
  GraphicText(handle,125,45,line1);
  GraphicText(handle,125,75,line2);
  SetAbsCharHeight(handle,8,i,i,i);
  SetScreenBase(screen[0],screen[0],1);
  SetScreenBase(screen[2],screen[2],1);
  Subliminal;
  Swirl; sc:=2;
  END;
END Part25.

```



Prologomenon

by Joseph Schmuller

In the last article, we saw how variables could be instantiated to important words in sentences recognized by a Definite Clause Grammar (DCG); then we saw that these instantiations could be asserted as facts into Prolog's database. At the end of the article, we showed that a DCG could accommodate a list of adjectives. Listing 1 (all listings are at the end of the article) is a DCG which summarizes these capabilities.

All of this leads naturally to the idea that whole phrases (containing determiners and adjectives) -- rather than just individual words -- should be instantiated to variables. That is, given the sentences:

That boy chases this little dog.

and:

This tiny little boy chases that big huge dog.

Instead of setting up our instantiations so that we assert:

```
chases(boy,dog)
```

for both, let's organize our DCG so that we assert:

```
chases([that,boy],[this,[little],dog]).
```

for the first sentence and:

```
chases([this,[tiny,little],boy],[that,[big,huge],dog]).
```

for the second. Notice that the adjectives reside in sub-lists within the phrase-lists. This, of course, is because of the way "descriptors" works. Listing 2 shows that if we make one simple change to the arguments of the "noun_phrase" rewrite rule, we can do the trick. The change is to have the arguments reflect the structure of the noun-phrase, and to put them in a list. Then, when a noun-phrase is found, this list gets unified to the argument for "noun_phrase" on the right-hand side of the "sentence" rewrite rule. Such a list will also get unified to the argument for "noun_phrase" on the right-hand side of the "verb_phrase" rewrite rule.

Parse Trees

Everything we've done thus far has been geared toward using DCGs to neatly turn English sentences into facts which can be asserted into Prolog's database, and then used by Prolog.

There is another use of DCGs, however. As you may recall from your earliest scholastic encounters with "grammar", there is a procedure called "diagramming sentences". This procedure results in a picture which illustrates the structure of a sentence, and the picture is

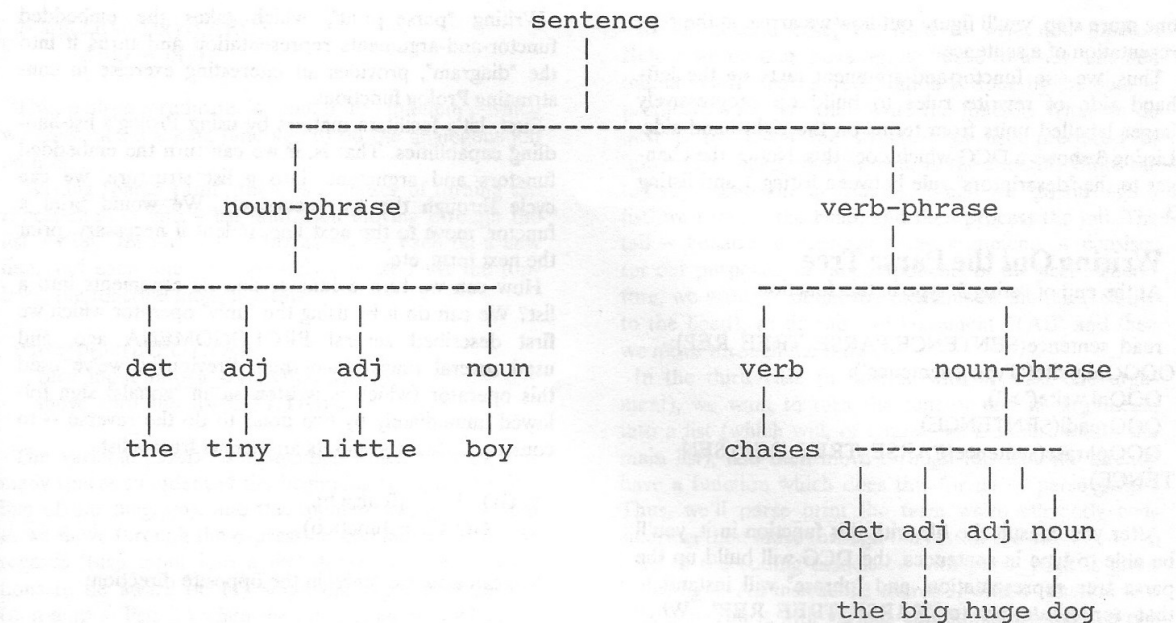


figure 1- Parse Tree

called a "parse tree". Figure 1 illustrates a parse tree for the sentence:

The tiny little boy chases the big huge dog.

Notice that in addition to grouping the words into phrases, each individual word is given its respective part of speech as a label. Linguists use diagrams like this one when they study sentence structure, and you should, too, if you want to explore computational linguistics.

We can set up a DCG so that a data-structure which represents a parse tree will result when we apply the built-in predicate "phrase" to a sentence. That is:

```
?-phrase(sentence(X),[the,boy,chases,the,dog]).
```

will give us:

```
X = sent(np(det(the),n(boy),
            vp(v(chases),np(det(the),n(dog))))).
```

In order to accomplish this, we have to be very clever about the arguments we provide to the terms in the DCG. So far, we've made sure that each term has had one argument by making that argument be a list. There is another way to do this, however. We can have the argument to a term like "verb phrase" be a Prolog fact as in:

```
verb_phrase(vp(V,N)) --> verb(v),noun_phrase(N).
```

The Prolog fact "vp(V,N)" reflects the structure of the verb-phrase in much the same way that the lists of arguments reflect structure within listing 2, but these two techniques are used for different purposes. The Prolog fact provides a mechanism for labelling the components of the verb-phrase; the previously-used argument-lists provide a convenient method for asserting things into the database. Anyway, if this verb-phrase rewrite rule is in the DCG, then when the DCG recognizes a pattern in the input which conforms to the right-hand side of this rule, the instantiations of V and N become instantiated to the arguments of the functor "vp" in the argument of "verb_phrase". The functors of facts like this will eventually be the labels of the parse tree; the arguments of these functors will be the components for which the functor is the label.

To make these points a little clearer, let's look at a DCG from the bottom up. We can represent the rewrite rules for two typical terminals as:

```
noun(n(dog))-->[dog].
```

```
determiner(det(the))-->[the].
```

If we specify one rewrite rule for "noun_phrase" as:

```
noun_phrase(np(D,N))-->determiner(D),noun(D).
```

then the DCG can instantiate "det(the)" to "D" and "n(dog)" to "N" on the right-hand side of the rule. It follows that "np(det(the),n(dog))" will be the instantiation on the left-hand side. If you extrapolate this procedure

one more step, you'll figure out how we arrive at the representation of a sentence.

Thus, we use functor-and-argument facts on the left-hand side of rewrite rules to build up progressively larger labelled units from terms on the right-hand side. Listing 3 shows a DCG which does this. Notice the changes to the "descriptors" rule between listing 1 and listing 3.

Writing Out the Parse Tree

At the end of listing 3, type in this function:

```
read_sentence(SENTENCE,PARSE_TREE_REP):-
OOOOnl,write('Type a sentence'),
OOOOnl,write('>'),
OOOreadl(SENTENCE),
OOOphrase(sentence(PARSE_TREE_REP),SENTENCE).
```

After you consult the file with this function in it, you'll be able to type in sentences, the DCG will build up the parse tree representation, and "phrase" will instantiate that representation to "PARSE_TREE_REP". When "PARSE_TREE_REP" is written out on the screen, however, it's in a format which isn't very easy on the eyes. What we need instead is a function which will take "PARSE_TREE_REP" and turn it into something prettier. The prettiest thing we could turn it into, of course, would be a picture like the one in figure 1, but a function to do this would be a pretty big job. Instead, we can imagine a diagram like that one laid on its side. To make things easier for us, let's forget about the connecting lines and adopt the convention that if term C is a subcomponent of term B, then C will appear below B and indented relative to B; if C is below B but not indented, then we will understand that B and C are at the same level and are subcomponents of a third term, term A. For example, the data-structure:

```
sent(np(det(the),n(boy)),vp(v(chases),np(det(the),n(dog))))
```

would be diagrammed as:

```
sent
  np
    det
      the
    n
      boy
  vp
    v
      chases
    np
      det
        the
      n
        dog
```

Writing "parse_print", which takes the embedded functor-and-arguments representation and turns it into the "diagram", provides an interesting exercise in constructing Prolog functions.

First, let's facilitate matters by using Prolog's list-handling capabilities. That is, if we can turn the embedded functors and arguments into a list structure, we can cycle through the list recursively. We would print a functor, move to the next line, indent if necessary, print the next term, etc.

How can we turn a functor and its arguments into a list? We can do it by using the "univ" operator which we first described several PROLOGOMENA ago, and used several times since then. Previously, we've used this operator (which is written as an "equals" sign followed immediately by two dots) to do the reverse -- to construct a functor and its arguments from a list.

```
?- GOAL =.. [func,a,b].
GOAL = func(a,b)
```

We can also use "univ" in the opposite direction:

```
?- func(a,b) =.. LIST.
LIST = [func,a,b]
```

```
?- func(a(x),b(y)) =.. LIST.
LIST = [func,a(x),b(y)]
```

Just as we would have expected, the functor becomes the head of the list and the arguments become the tail.

Notice that in the last example, elements a(x) and b(y) each remain as a functor and an argument. In parse_print, after we first apply "univ", as we cycle through the list, we will apply "univ" again whenever we come to an element which is a functor with at least one argument. When we come to a functor with no arguments (i. e., a functor whose arity is zero), we'll move to a new line and write the functor. Bear in mind that the only time we ever write a term is if that term has an arity of zero. How can we test to see if a functor has any arguments or not? A built-in predicate called "functor" does the job:

```
?-functor(f(a(x),b(y)),FUNCTOR,ARITY).
FUNCTOR = f ARITY = 2
```

As you can see, "functor" has three arguments. The first is the expression to be evaluated, the second is a variable which gets instantiated to the functor of the expression, and the third is a variable which gets instantiated to the arity of the expression. By the way, there's a companion predicate called "arg" (which we won't be using) that instantiates a variable to a specified argument of an expression:

```
?- arg(1,f(a(x),b(y)),ARG).
ARG = a(x)
```

```
?- arg(2,f(a(x),b(y)),ARG).
   ARG = b(y)
```

This built-in predicate is similar to "nthelem", which we constructed a long time ago to do the analogous job on lists.

Writing "parse_print", then, is a matter of turning the representation into a list, and then moving through the list, writing zero-arity functors as we go, each on a new line, and each one appropriately indented. At the top-level, the function looks like this:

```
parse_print(INPUT,TAB) :-
  turn_input_into_a_list(INPUT,LIST),
  move_thru_the_list(LIST,TAB).
```

The variable "TAB" is a number which tells us how many spaces to indent at the beginning (i. e., on the first line of our diagram), and this number will get updated as we move through the expression and indent terms. As regards "turn_input_into_a_list", there are three conditions to be aware of: (1) when the input is already a (non-empty) list; (2) when the input is an empty list; (3) when the input is a functor with its arguments. The first condition will occur when we come to the last element of a list. That last element constitutes the remainder of the tail of the list, and because it's a tail, it is itself a list. If we try to apply "univ" to this list (or to any other), we'll get an interesting result. Try putting a list like [a,b,c,d] on the left-hand side of "univ" and a variable on the right-hand side and you'll see this for yourself. The second condition will occur after we've finished going through a list. The third condition is the one which is likely to occur most often. Here's how we deal with the three conditions:

```
turn_input_into_a_list([H|T],[H|T]) :-
  [H|T] \= [].
turn_input_into_a_list([],_).
turn_input_into_a_list(INPUT,LIST) :-
  INPUT =.. LIST.
```

In the first case we leave the list alone; in the second case, we don't bother instantiating anything; and in the third case, we apply "univ".

With respect to "move_thru_the_list", there are also three conditions that we might encounter as we process each of the elements in a list: (1) when we're trying to move through an empty list; (2) when we've come to a functor with no arguments; (3) when we come to a functor that has at least one argument.

In the first case, which occurs when we've finished going through a list, we just replace the second argument with the anonymous variable:

```
move_thru_the_list([],_).
```

In the second case, we want to write the functor. Before we do that, however, we move to a new line and indent "TAB" spaces. Indentation is done by the built-in predicate "tab". We then write the functor. What we do next is based on the structure of lists produced by "univ". The head, you'll recall, is the functor, while the tail represents the arguments. When we "go through a list" we process the head, and then process the tail. The tail -- because it represents the arguments -- consists, for our purposes, of subcomponents of the head. Therefore, we want to indent the elements in the tail (relative to the head); to do this, we increment "TAB" and then we move through the tail.

In the third case (a functor with at least one argument), we want to turn the functor and its arguments into a list (which will, of course, be a sublist within the main list), and then move through this list. We already have a function which does this for us -- "parse_print"! Thus, we'll parse_print the term we're currently considering, then move through the rest of the list.

All of the coding associated with parse_print is given in listing 4. An instructive exercise, with a function like this, is to run it with various conditions commented out (such as the empty list conditions); try it and see what happens. If you add parse_print to read_sentence, you'll have a function which allows you to type in a sentence, and then returns it parsed, labelled, and diagrammed. We've set up parse_print specifically to deal with the structures that result from parsing sentences. These structures always have functors with an arity of one at the deepest level. This is because the deepest level represents one word labelled with its part of speech. Structures that have functors with arity greater than one at their deepest levels will not be diagrammed appropriately by parse_print -- the indentations which correspond to the deepest level will be a little out of whack.

I hope you've had an enjoyable summer. See you next month.

MORE 

Advertiser Index

page:		
BC	Avant-Garde Systems	(904) 221-2904
13	Proco Products	(800) 843-1223
45	Berkeley Microsystems	(415) 465-6956
45	Softworks West	
73	Softrek	(407) 657-4611

```

/*****Listing 1-- A DCG with instantiation*****/
/*****that represents a parse tree*****/
/*****Listing 2-- A DCG with instantiation*****/
/*****that represents a parse tree*****/

sentence((SUBJ, VERB, OBJ))
--> noun_phrase(SUBJ), verb_phrase(VERB, OBJ).
noun_phrase(NOUN) --> det(DET), noun(NOUN).
--> det(DET), descriptors(ADJ_LIST), noun(NOUN).
verb_phrase(VERB, NOUN) --> verb(VERB), noun_phrase(NOUN).
verb(chases) --> [chases].
noun(boy) --> [boy].
noun(dog) --> [dog].
det(the) --> [the].
det(this) --> [this].
det(that) --> [that].
adjective(big) --> [big].
adjective(huge) --> [huge].
adjective(tiny) --> [tiny].
adjective(little) --> [little].
adjective(small) --> [small].
descriptors((FIRST_ADJECTIVE|REST_OF_ADJS)) -->
adjective(FIRST_ADJECTIVE),
descriptors(REST_OF_ADJS).
descriptors((FIRST_ADJECTIVE)) -->
adjective(FIRST_ADJECTIVE).
adjective(FIRST_ADJECTIVE) --> [FIRST_ADJECTIVE].
/*****Listing 3-- A DCG that builds a data-structure*****/
/*****that represents a parse tree*****/
/*****Listing 4-- A DCG that builds a data-structure*****/
/*****that represents a parse tree*****/

sentence(sent(NP, VP))
--> noun_phrase(NP), verb_phrase(VP).
noun_phrase(np(det, ADJ_LIST, NOUN)) --> determiner(DET), noun(NOUN).
--> determiner(DET), descriptors(ADJ_LIST), noun(NOUN).
verb_phrase(vp(verb, NP)) --> verb(VERB), noun_phrase(NP).
verb(v(chases)) --> [chases].
noun(n(boy)) --> [boy].
noun(n(dog)) --> [dog].
determiner(det(the)) --> [the].
determiner(det(this)) --> [this].
determiner(det(that)) --> [that].
adjective(adj(big)) --> [big].
adjective(adj(huge)) --> [huge].
adjective(adj(tiny)) --> [tiny].
adjective(adj(little)) --> [little].
adjective(adj(small)) --> [small].
descriptors((adj(ADJ)|REST_OF_ADJS)) -->
adjective(adj(ADJ)),
descriptors(REST_OF_ADJS).
descriptors((adj(ADJ))) -->
adjective(adj(ADJ)).
adjective(adj(ADJ)) --> [adj(ADJ)].
/*****Listing 5-- A DCG that builds a data-structure*****/
/*****that represents a parse tree*****/
/*****Listing 6-- A DCG that builds a data-structure*****/
/*****that represents a parse tree*****/

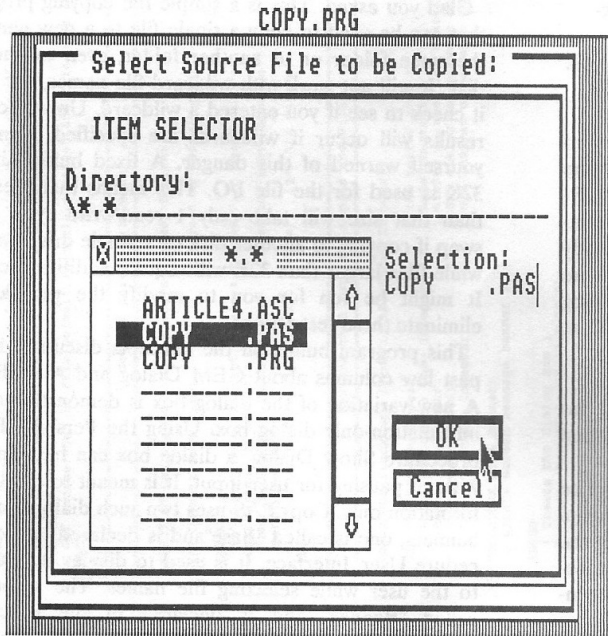
parse_print(INPUT, TAB) :-
turn_input_into_a_list(INPUT, LIST),
move_thru_the_list(LIST, TAB).
turn_input_into_a_list([H|_], [H|_]) :-
turn_input_into_a_list([], []).
turn_input_into_a_list([], []) :-
INPUT = [].
move_thru_the_list([], []).
move_thru_the_list([H|_], TAB) :-
turn_input_into_a_list([H|_], TAB),
move_thru_the_list([], TAB).
move_thru_the_list([], TAB) :-
TAB = 0,
write(FUNCTION, ARITY),
TAB is TAB + 1,
move_thru_the_list([], TAB).
move_thru_the_list([H|_], TAB) :-
turn_input_into_a_list([H|_], TAB),
move_thru_the_list([], TAB).
move_thru_the_list([], TAB) :-
TAB = 0,
write(FUNCTION, ARITY),
TAB is TAB + 1,
move_thru_the_list([], TAB).

```

END



STips.PAS



For the past several months, we've been covering the topics of GEM Alert and Dialog Boxes. This month we're going to add another skill to your programmer's toolkit, that of calls to GEMDOS. The program will demonstrate the use of these calls to copy a single file. It is not meant to be the answer to all of your copying needs, but serves the purpose of instruction well. As in previous columns, Personal Pascal, ver 2.02 is being utilized.

What are GEMDOS Calls ?

Operating systems are responsible for managing hardware resources and file access. As such, they have their own routines to accomplish these actions. These functions are generally independent of the hardware being used. Typically, programmers can access these routines via function calls from a high-level programming language such as Pascal and C. Many of the routines have already been duplicated in the language itself. For example:

Pascal's RESET(FileVar,FileName);

Opens a file for read access as does a call to

```
FUNCTION      TOS_Open      (VAR
FileName:Path;Mode:Short_Integer):Short_Integer;
```

by Timothy S. Magee
Compulogic Development Corp.

GEMDOS (\$3D);

Some routines, however, are rarely duplicated in a high-level language, such as those that manage memory access or read characters from the keyboard.

BIOS and XBIOS Too !

In addition to the operating system calls, personal computers have additional built-in routines that are specific to that machine called the BIOS (Basic Input Output System) and often XBIOS for eXtended BIOS. As the names suggest, these routines are concerned with fundamental input/output functions. Although our example program does not contain any BIOS or XBIOS calls, they are performed in a similar manner as the GEMDOS calls.

Prerequisites

Before you can properly use any of these functions, you need a few reference tools at your disposal:

- 1) A compiler that supports calls to GEMDOS such as Personal Pascal.
- 2) A reference manual of GEMDOS, BIOS, & XBIOS calls. Two good sources are Atari ST Internals by Abacus Software and the Atari Developer's Toolkit from Atari Corp.

MORE

- 3) A third, highly recommended, source is to purchase Tackle Box ST from SRM Enterprises, also available through Jefferson Software. This package contains routines for all of the GEMDOS calls and documents those calls thoroughly, as well.

General Syntax for System Calls

Depending on the reference source you are using to describe the system calls, you may have to determine whether the call is a "FUNCTION" or a "PROCEDURE". Tackle Box tells you directly the type of declaration needed, in fact, the declarations are already made in an "include" file with the package. Most of the time, the calls are declared as FUNCTIONS, returning a resultant value or an error code. Let's take a look at an example from the program:

```
FUNCTION   TOS_Close (Handle: Short_Integer):Short_Integer; GEMDOS ($3E) { close an open file }
```

This is the GEMDOS function to close a file. Notice that to make the call, the parameters are passed, then a GEMDOS function is specified by simply indicating the function number as a parameter to the GEMDOS procedure. Since it is a function, this call returns an integer indicating any error that may have occurred while closing the file. Normally, the function number is given in hexadecimal format since it is usually described that way in most documentation. BIOS and XBIOS calls are performed the same way, substituting BIOS or XBIOS for GEMDOS as appropriate.

A Special Note about String Parameters

Since a lot of GEMDOS was written in 'C', the system calls expect any string parameters to be passed in the 'C' format. A string can be thought of as a Packed Array [1..MaxLen] of Char with some declared maximum length. The Personal Pascal string uses the first position of the array, position 0, to hold an ASCII character indicating the current length of the string. No special ending character is needed (Turbo Pascal also uses this same format). The 'C' string is a Packed Array of Char with the ending character being a NULL or CHR(0). Therefore, Pascal strings must be converted to 'C' strings when being passed as parameters to system calls. Fortunately, Personal Pascal provides a procedure, declared in AUXSUBS.PAS, to perform just such a maneuver. The syntax of the procedure call is:

```
P_to_Cstr (PasStr, CStr);
```

where: PasStr is a Pascal String to be converted
and : CStr is the converted 'C' string which is really a Packed Array [1..255] of Char.

How Does the Example Program Copy.PAS Work?

Glad you asked. This is a simple file copying program that can be used to copy a single file to a new name in the same folder or to another folder, even to another disk. It will not work with wildcard file names, nor does it check to see if you entered a wildcard. Unpredictable results will occur if wildcards are specified. Consider yourself warned of this danger. A fixed buffer size of 32K is used for the file I/O. This means that files less than that size will take only 1 read/write cycle (disk swap if copying to another disk on a single drive system) while files larger than 32K will require additional cycles. It might be fun for you to modify the program to eliminate these restrictions.

This program builds on the concepts discussed in the past few columns about GEM Dialog and Alert Boxes. A new variation of the dialog box is demonstrated: the information-only dialog box. Using the Personal Pascal procedure Show_Dialog, a dialog box can be displayed without pausing for user input. It is meant to display information only. Copy.PAS uses two such dialog boxes or banners, one is called "Base" and is declared in the procedure User_Interface. It is used to display instructions to the user while selecting file names. The second is called "Banner" and is declared in the procedure BlockCopy. An in-process copy message is displayed in this banner during the actual file copying process. The syntax for the Show_Dialog call is as follows:

```
Show_Dialog (Box,Index);
```

where: Box is the GEM dialog pointer from call to New_Dialog

and : Index is the index number of the item to be displayed, use an index of 0 to display the entire dialog box.

The pre-defined dialog box, Get_In_File, is used to retrieve both the source file name and the destination file name. Alert boxes are used to display all error messages and confirmations. Immediately following the variable declaration section are the GEMDOS declarations. Otherwise, the source code is fully commented and should explain the more conventional Pascal commands.

Stay Tuned for GEM Menus

Next month, we will discuss GEM Menus. Your programs can create and use their own pull-down menus, just like the one on the desktop. We've also begun to evaluate another implementation of Pascal for the ST, Prospero Pascal, and hope to have some initial thoughts on it next month as well. Until then, good computing.

```

Program Name.....: Copy.PAS ver 1.1
Author(s).....: Tim Magee
Date/Rev Date.....: 07/22/88
Description.....: Simple file copying utility program.
Notice.....: Released to the Public Domain 1988.
              by Compulogic Development Corp., Inc.
              This program is for demo purposes only.
              NO warranty, explicit or implied, applies!
              You MAY use, alter, redistribute it freely.

This program demonstrates the use of GEMDOS calls from Personal Pascal 2.02..

PROGRAM Copy ;
( Include Pascal Routines)

($I GEMSubs.PAS)
($I AUXSubs.PAS)

CONST BlockSize=MaxInt;
      WriteMode=1;
      ReadMode=0;
      Attrb=0;

TYPE BufType=array [1..BlockSize] of Byte; ( define buffer type)
      CopyFile=FILE of BufType; ( a file )

VAR SourceTargetCopyFile; ( pascal file variables)
      ToFile,FromFile:Path Name; ( file name string )
      SrcFile,TargetFile:C-String; ( file name C-style string)
      Success,FromFile,ToHandle:Short_Integer; ( GEMDOS file handles)
      BaseDialog_Ptr; ( main dialog box indices)
      Instruct:Short_Integer;
      AlertStr:String(255); ( Alert Box message )
      Fatal:Short_Integer; ( error ID, Alert Box Choice)
      Fatal:Boolean; ( cannot go on flag )

( =====
  GEMDOS Disk/File Calls
  ===== )

FUNCTION TOS_Create (VAR Name:C-String; Attrb:Short_Integer):Short_Integer;
GEMDOS ($35);
( create a new file )

FUNCTION TOS_Open (VAR Name:C-String; Mode:Short_Integer):Short_Integer;
GEMDOS ($3d);
( Open an existing file )

FUNCTION TOS_Close (Handle:Short_Integer):Short_Integer;
GEMDOS ($3e);
( Close a file )

FUNCTION BlockRead (Handle:Short_Integer; ByteCount:Long_Integer;
VAR Buffr:BufType): Long_Integer;
GEMDOS ($3f);
( Read a block from file )

FUNCTION BlockWrite (Handle:Short_Integer; ByteCount:Long_Integer;
VAR Buffr:BufType): Long_Integer;
GEMDOS ($40);
( Write a block to file )

( =====
  Create a User Dialog Interface
  ===== )

PROCEDURE User_Interface;
BEGIN
  Base:= New Dialog(5,19,2,43,22);
  Set_Dialog(Base,Dialog);
  Set_DText(Base,Instruct,'System_Font,TE_Center);
END;

( =====
  Get The Source File Name
  ===== )

FUNCTION Get_Source (VAR FromFile:Path Name):Boolean;
VAR Default:Path Name;
( passed as default path )

```

```

BEGIN
  Set_DText(Base,Instruct,' Select Source File to be Copied:
System_Font,TE_Center);
  Show Dialog(Base,Instruct); ( show the above message )
  Default:='*. *';
  Get_Source:=Get_In_File(Default,FromFile); ( get the file name in FromFile)
END;

( =====
  Get the Target File Name
  ===== )

FUNCTION Get_Target (VAR ToFile:Path Name):Boolean;
VAR Default:Path Name;
( passed as default path )
BEGIN
  Set_DText(Base,Instruct,' Select Target File Destination: '
System_Font,TE_Center);
  Show Dialog(Base,Instruct); ( show the above message )
  Default:='*. *';
  Get_Target:=Get_In_File(Default,ToFile); ( get the filename in ToFile )
END;

( =====
  Routine that Actually Copies the File
  ===== )

PROCEDURE BlockCopy (FromFl,ToFl:Short_Integer);
VAR IOBuffer:BufType; ( the buffer area )
      ActualWritten : Long_Integer; ( byte counters )
      Instruct:Short_Integer; ( dialog pointer for message )
      Message:Short_Integer; ( GEMDOS file handles )
      GenMsg : Short_Integer;
      GenMsg : Short_Integer; ( GEM indices of message items )

PROCEDURE Make_Banner;
( create screen message banner )
BEGIN
  Create the Dialog Box )
  Banner:= New Dialog(5,13,8,50,9);
  ( Display a Title Line )
  Banner:= Banner;
  GenMsg:= Add_Dialog_Banner,GenMsg,MsgStr,System_Font,TE_Center);
  Set_DText(Banner,GenMsg,MsgStr,System_Font,TE_Center);
END; ( Make_Banner )

BEGIN (Main Routine of BlockCopy)
  Make_Banner;
  Center_Dialog(Banner);
  Show Dialog(Banner,0);
  REPEAT
    Actual:=BlockRead (FromFl,BlockSize,IOBuffer); ( read a block from source)
    IF Actual>0 THEN BEGIN
      Write:=BlockWrite (ToFl,Actual,IOBuffer); ( write block to target)
      IF Write=<Actual THEN BEGIN
        Fatal:=true;
        FatalCode:=10;
        Actual:=0;
        END; (IF Written)
      END; (IF Actual>0)
      IF Actual<0 THEN BEGIN
        Fatal:=true;
        FatalCode:=11;
        UNTIL Actual<=0;
        End Dialog(Banner);
      END; (BlockCopy)
    )
  ( =====
    A Function to determine existence of a file
    ===== )

FUNCTION Exist (FileName:Path Name):Boolean;
VAR f:FILE of Byte;

```



```

( =====
Main Program Routine
===== )

BEGIN (Main Program)
  IF Fatal=0 THEN BEGIN
    Init Mouse;
    Hide Mouse;
    User Interface;
    Show Dialog(Base,0);
    IF Get_Source(FrontFile) THEN BEGIN
      End Dialog(Base);
      P TO Cstr (FromFile,SrcFile);
      IF EXIST(FromFile) THEN BEGIN
        IF FrontHandle<0 THEN BEGIN
          ErrorCode=7;
          Fatal:=true;
        END;
      END;
    END;
  END;
  IF Get_Target(TopFile) THEN BEGIN
    P TO Cstr (TopFile,TargetFile);
    IF EXIST(TopFile) THEN BEGIN
      Error(13);
      IF Dummy=2 THEN BEGIN
        ToHandle:=TOS_Open (TargetFile,WriteMode);
        IF ToHandle<0 THEN BEGIN
          ErrorCode=8;
          Fatal:=true;
        END;
      END;
    END;
  END;
  ELSE BEGIN
    Fatal:=true;
    ( user chose NOT to overwrite )
    ( cannot go on )
    ( assign error message )
  END;
  ELSE BEGIN
    ToHandle:=TOS_Create (TargetFile,Attrib);
    IF ToHandle<0 THEN BEGIN
      ErrorCode=9;
      Fatal:=true;
      ( Target file does not already exist )
      ( create target file )
      ( check for create error )
      ( cannot go on )
    END;
  END;
  IF NOT Fatal THEN BlockCopy(FromHandle,ToHandle);
  ELSE BEGIN
    End Dialog(Base);
    Fatal:=true;
    ErrorCode=2;
    ( i.e. Get_Target is false )
    ( Destination file not specified )
  END;
  ELSE BEGIN
    Fatal:=true;
    ErrorCode=3;
    ( Source file does not exist )
    ( assign error code )
  END;
  ELSE BEGIN
    End Dialog(Base);
    Fatal:=true;
    ErrorCode=4;
    ( i.e. Get_Source is false )
    ( Source file not specified )
  END;
  IF NOT Fatal THEN ErrorCode:=12;
  IF NOT Fatal THEN BEGIN
    Exit_GEM;
    ( display error message )
    ( tell GEM we're done )
  END;
END. (program copy)

```

END

```

( =====
Error Handler
===== )

PROCEDURE Error (ErrorCode:Byte);
TYPE Valid = 1..13;
Codes = set of Valid;
VAR CloseSrc,CloseTarg : Codes;

BEGIN
  CloseSrc := {1,2,8..11};
  CloseTarg := {10..12};
  AlertStr:='[3]';
  CASE ErrorCode OF
    1 : AlertStr:=Concat(AlertStr,['file Will NOT Be Overwritten !'],
      '{ ABORT COPY }');
    2 : AlertStr:=Concat(AlertStr,['Target File NOT Specified !'],
      '{ ABORT COPY }');
    3 : AlertStr:=Concat(AlertStr,['FromFile, "" Cannot Be Found !'],
      '{ ABORT COPY }');
    4 : AlertStr:=Concat(AlertStr,['Source File NOT Specified !'],
      '{ ABORT COPY }');
    5 : AlertStr:='[1][Error Closing Source File ][ Check file ]';
    6 : AlertStr:='[1][Error Closing Target File ][ Check file ]';
    7 : AlertStr:=Concat(AlertStr,['Error Opening Source File !'],
      '{ ABORT COPY }');
    8 : AlertStr:=Concat(AlertStr,['Error Opening Target File !'],
      '{ ABORT COPY }');
    9 : AlertStr:=Concat(AlertStr,['Error Creating Target File !'],
      '{ ABORT COPY }');
    10 : AlertStr:=Concat(AlertStr,['Error Writing to Target File !'],
      '{ ABORT COPY }');
    11 : AlertStr:=Concat(AlertStr,['Error Reading from Source File !'],
      '{ ABORT COPY }');
    12 : AlertStr:='[1][Copy Has Been Completed.][ Good Job ]';
    13 : AlertStr:=Concat('12][Overwrite Existing "',TopFile,'" ?]',
      '{ NO | YES }');
  END;
  IF ErrorCode IN CloseSrc THEN BEGIN
    CloseSrc:=TOS_Close(FromHandle);
    IF Success<>0 THEN Error(5);
  END;
  IF ErrorCode IN CloseTarg THEN BEGIN
    Success:=TOS_Close(ToHandle);
    IF Success<>0 THEN Error(6);
  END;
  Dummy:=Do Alert(AlertStr,1);
END; (Error)

```



Pennsylvania 6-8000

by Eric Thornton

"Mumbling Mnemonics II"

The rest of mnemonics are just as simple to remember as the first half presented in the June issue.

JMP JuMP to address
JSR Jump to SubRoutine
LEA Load Effective Address
LINK LINK temporary stack storage space
LSL Logical Shift Left
LSR Logical Shift Right
MOVE MOVE data in memory
MULS MULTiply Signed data
MULU MULTiply Unsigned data
NBCD Negate Binary Coded Decimal
NEG NEGate standard number
NOP perform No OPERATION
NOT is this NOT something?
OR OR binarily
PEA Push Effective Address onto stack
RESET RESET all external devices
ROL ROTate Left bits

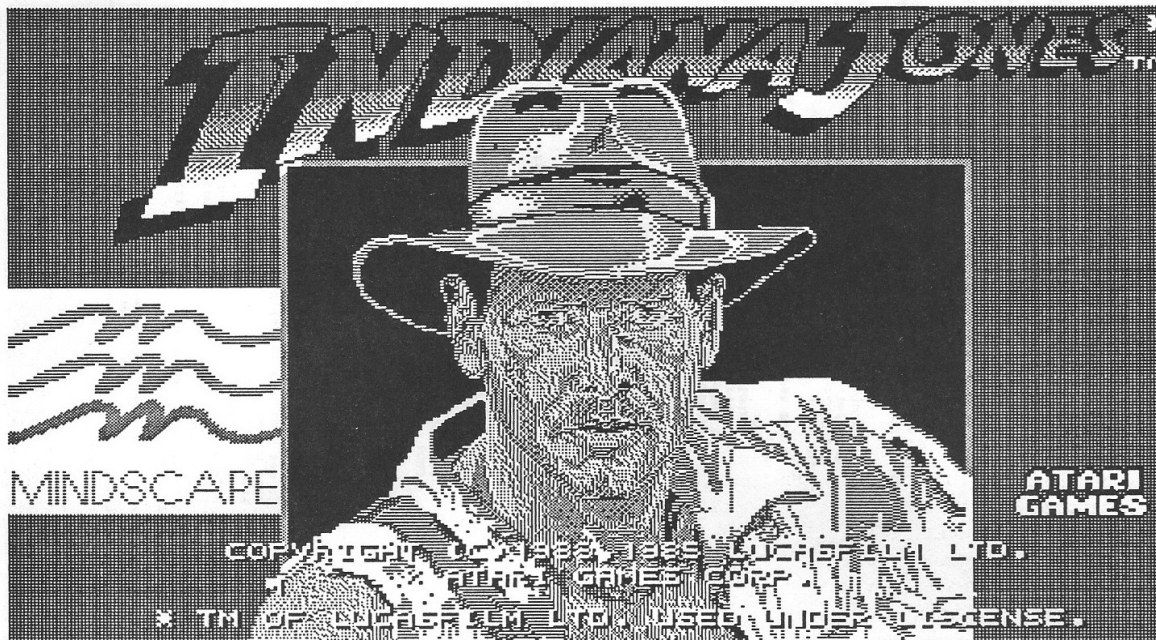
ROR ROTate Right bits
RTE ReTurn from Exeception
RTR ReTurn and Restore condition and stack
RTS ReTurn from Subroutine
SBCD Subtract Binary Coded Decimal
Scc Set bits on condition
STOP STOP and wait for interrupts
SUB SUBtract number
SWAP SWAP contents of two data registers
TAS Test And Set bits in data
TRAP TRAP stack and call supervisor mode routine
TST TeST for negative or zero
UNLK UNLinK stack frame allocated by LINK

Upcomming topics include GEMDOS, BIOS, XBIOS, VDI and AES calls and calling formats; an auto folder program to set screen colors to PANEL defaults without wasting all that memory and a command line parser for chopping up input lines and picking out the useful information. Until next time, enjoy.

END



Review Board



INDIANA JONES and the **TEMPLE OF DOOM**[™]

by U.S. Gold Limited
distributed by
Mindscape, Inc.
3444 Dundee Road
Northbrook, IL, 60062.
(312) 480-7667
\$44.95.

Indiana Jones and The Temple of Doom is a new arcade action game from *Mindscape*, the publishers of *Gauntlet*. The game was designed to emulate the popular arcade version produced by Atari's games division under license from Lucasfilm.

If you have seen the movie of the same name, you know the basic theme behind this game. Indiana Jones must find and release a group of imprisoned children while searching for three missing Shankara Stones. Along the way he must climb ladders, swing across gorges, and traverse lava filled pits while dodging bats and snakes. At the same time Indy is being chased by a fireball throwing priest and his randomly appearing Thugee henchmen.

The only way to fight these foes is with a bullwhip. Enemies can be briefly whipped into submission or, in some cases, destroyed using this versatile weapon. The bullwhip is also used to break the locks on prisoners cages and aid Indy in swinging Tarzan style across gaps in mountain trails.

After completing the first stage Indiana Jones must traverse a brief maze of crumbling railroad tracks in a mine car. At the same time, similar mine cars full of Thugees are attempting to derail the car in which your character is riding. Challenges include balancing across damaged car rails and avoiding destruction on dead end railways. After having once again eluded death, the final test comes when Indy must cross a trap door above a lava pit and rescue a waiting Shankara Stone. Having accomplished this goal, you have completed the level.

Subsequent screens repeat the basic theme with increased complexity. More prisoners must be rescued; additional Thugee villains must be avoided or whipped into temporary immobility; and more ladders and plateaus must be traversed to again reach the railroad car and Shankara Stones.

At the beginning of each game you can select one or two players, one of three difficulty levels, and whether or not you want to listen to music (a lively theme song from the movie). Players use the joystick to control

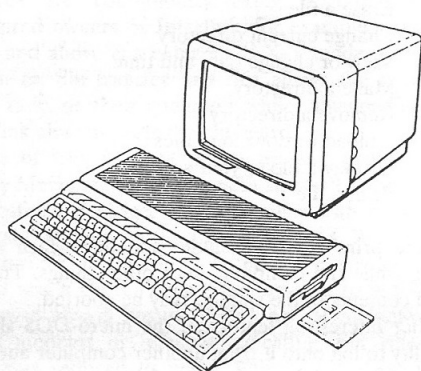
character movements and the fire button controls bullwhip actions.

Having played the arcade game, I must admit to being disappointed with the ST version of **Indiana Jones and The Temple of Doom**. The arcade version featured quick scrolling graphics of a complexity that could have been emulated on the ST. The home version uses slow scrolling graphics that would have looked more suitable on an 8 bit system. It was not uncommon to wait while the graphics scrolling attempted to "catch up" with the character. The arcade version used digitized voices and sounds from the movie. The home version plays a repetitive **Music Studio** quality theme through the speaker and uses sounds that are a grave injustice to the ST's sound chip.

I started by playing the easy level so I could learn the game. On my third time out I inadvertently entered the hard level and was able to clear all stages and finish the game using only three of my characters five lives. I rapidly grew bored with **Indiana Jones and The Temple of Doom**. I found the game lacked enough variation to keep play interesting. With the exception of my first time maneuvering a mine car, the game presented no real challenges.

Mindscape has produced other excellent ST games. I was sincerely disappointed that the programmers assigned this ST conversion didn't take more time and effectively use the capabilities of the system. This game should be enjoyed by children and pre-teens. However, I believe adults with a yen for arcade action should avoid **Indiana Jones and The Temple of Doom**.

reviewed by Dan Fruchey



ST Applications

The Atari ST Journal

P.O. Box 980
Forestville, CA 95436

☐ \$4.50/ Single issue ☐ \$20/ 6 issues ☐ \$32.50/ yr.

☐ \$12/ Single issue ☐ \$50/ 6 issues ☐ \$92.50/ yr.

☐ Renewal ☐ Start subscription with _____ issue

Name: _____

Address: _____

City: _____ State: _____ Zip: _____

Foreign subscriptions are \$40 per year (\$78 for Air Mail).
Make payment with check on US Bank or Postal Money Order

ST Applications

The Atari ST Journal

P.O. Box 980
Forestville, CA 95436

☐ \$4.50/ Single issue ☐ \$20/ 6 issues ☐ \$32.50/ yr.

☐ \$12/ Single issue ☐ \$50/ 6 issues ☐ \$92.50/ yr.

☐ Renewal ☐ Start subscription with _____ issue

Name: _____

Address: _____

City: _____ State: _____ Zip: _____

Foreign subscriptions are \$40 per year (\$78 for Air Mail).
Make payment with check on US Bank or Postal Money Order

CONNECT

STT Systems
P.O. Box 2315
Titusville, FL. 32796
(305) 383-0353
\$29.95



Anyone who has used Chat by SST systems will immediately recognize the similarities to **Connect**, SST system's new terminal program. (See my January 1987 **ST Applications** article for a review of Chat).

Like Chat, **Connect** cannot be configured without exiting and running a separate configuration program. However, it is now possible to edit the user-defined function keys. It is still necessary to hold down shift to access the ten function keys, and they cannot be changed from within **Connect**. The dialer phone numbers and port configuration are configured in the configuration program. As before, the port parameters are automatically set when a number is dialed. Unlike Chat, however, the port parameters may be changed without exiting the terminal program, thanks to the addition of the micro-DOS shell.

Perhaps the biggest change from Chat is the addition of the micro-DOS shell. This command line interpreter allows you to perform a number of maintenance operations from within **Connect**. As an ST user, you may be unfamiliar with command line interpreters. On computers without a graphics operating environment such as GEM, commands are typed into a program called a command line interpreter, or shell. While **Connect's** micro-DOS shell will not replace your GEM desktop, it will allow you to do basic disk operations and set the serial port configuration. Some other functions provided by the micro-DOS are:

- Display the file directory
- Copy a file
- Rename a file
- Erase a file
- Change current directory
- View or change date and time
- Make a directory
- Remove a directory
- Upload or download files
- Display a file to your screen
- Print a file

The file print feature includes spooling which allows printing while the computer does other things. The unprinted contents of the spooler may be aborted.

Another interesting feature of the micro-DOS shell is the ability to log onto it from another computer and control your ST remotely. However, it has no access control, so it is not a good idea to use this feature over a modem unless you don't mind the possibility of someone dialing in and viewing, downloading, or deleting all

Continued on page ==> 70

Please Answer Each of These Questions

1. This copy of ST Applications came to me through:
A. Friend C. User Group
B. Retail Store D. School
2. I personally use:
A. 520ST C. Double Sided Disk Drive
B. 1040ST D. Hard Disk Drive
3. I intend to purchase the following peripherals in the near future:
A. SS Disk Drive E. DS Disk Drive
B. Hard Disk Drive F. Video Digitizer
C. Audio Digitizer G. MIDI Synthesizer
D. Printer H. Modem
4. Which section do you read the most:
A. Articles B. Reviews
5. How many people will read this copy of ST Applications? _____
6. I'd like to see ST Applications devote more coverage to:
A. Games E. Graphics
B. Business F. Education
C. Utilities G. Tutorials
D. Other _____
7. I spent on software in the past 12 months:
A. less than \$100 D. \$600-\$1000
B. \$100-\$300 E. More than \$1000
C. \$300-\$600
8. I use the source listings in ST Applications:
A. Not at all C. Glance at them
B. Type them in D. Study them thoroughly
9. How many programs did you (will you) type in from this issue? _____
10. My favorite article in this issue is: _____

Please Answer Each of These Questions

1. This copy of ST Applications came to me through:
A. Friend C. User Group
B. Retail Store D. School
2. I personally use:
A. 520ST C. Double Sided Disk Drive
B. 1040ST D. Hard Disk Drive
3. I intend to purchase the following peripherals in the near future:
A. SS Disk Drive E. DS Disk Drive
B. Hard Disk Drive F. Video Digitizer
C. Audio Digitizer G. MIDI Synthesizer
D. Printer H. Modem
4. Which section do you read the most:
A. Articles B. Reviews
5. How many people will read this copy of ST Applications? _____
6. I'd like to see ST Applications devote more coverage to:
A. Games E. Graphics
B. Business F. Education
C. Utilities G. Tutorials
D. Other _____
7. I spent on software in the past 12 months:
A. less than \$100 D. \$600-\$1000
B. \$100-\$300 E. More than \$1000
C. \$300-\$600
8. I use the source listings in ST Applications:
A. Not at all C. Glance at them
B. Type them in D. Study them thoroughly
9. How many programs did you (will you) type in from this issue? _____
10. My favorite article in this issue is: _____

INTERLINK ST

Advanced Communications Made Easy

Intersect Software Corp.

3951 Sawyer Road, Suite 108

Sarasota FL 33583

800-826-0130

FL 813-923-8774

\$39.95

Inside *Intersect's Interlink* is all the telecommunication power anyone needs, and a whole lot more. Boasting the ease of use that a beginner can master and the sophistication to please the most demanding user, it lives up to its promises. Auto dial any BBS, and don't worry about remembering baud rates or log-on information. Disk commands, complete with 400K and 800K formatting smooth out the creation of disks and managing programs. Online help and connect chimes are niceties that add the little extra touch. XMODEM, XMODEM CRC, YMODEM, and ASCII file transfers to disk or capture buffer. With the continuous update of new protocols and emulation modes. *Interlink* will allow the running of programs from within the *Interlink* environment. Use of customized translation table with buffer window and editor let the user see and edit what has been received. Continuous status window keeps all information accessible. Record function allows the user to make recordings of all keyboard actions and make creating custom macros easy. Type ahead buffer makes it easy to chat online. Vt52, vt100 and user definable translation allow the user to pick his/hers telecommunication environment. 20 macro keys and built in password protection help to make *Interlink* a very powerful program. As if all this was not enough a complete auto answer mode allows the use of passwords and different levels of access to files. Call logging is as easy as turning it on. This allows the user to know exactly how much time was spent in all communications. Some features are continuously being upgraded free to registered owners of *Interlink*, for example: version 1.7 users and above can enter multi-X-Y mode in the (???) button on file transfer and this allows the user to do other tasks on their computer while downloading a file. *Interlink* also supports the clipboard.

Ease of use, but not giving up any power is what Randy Mears author of *Interlink* stressed about his communications package. I was familiar with communications at 300 baud on the Commodore 64. Before buying my Atari ST, just getting a successful communication of ASCII was a chore. Then transferring it to usable basic code was always unsure. I was not at all enthusiastic about modems or telecommunicating, but when 1200 baud was reduced to under \$200.00 and there was all this free software to download, I decided to get one. I quickly learned the use of a public domain communications package with XMODEM download protocol by Randy Mears. This was at least some what reliable, but was tedious to keep up with all the baud rate settings,

passwords, and other options. When *Interlink* first was being beta tested I was able to get a look at it. I was in shock at how easy it was for the intersect sect (those who had beta copies) to communicate over the modem (it made me sick to tell the truth). As soon as it was available for sale I purchased a copy and didn't have to worry about telecommunications again (until I had to switch from version .95 to version 1.8 to see all the new stuff to write this review). It was very easy to learn to use *Interlink*. Just point and click, everything is GEM orientated. With on-line help most information is available without a manual. I quickly had a telephone file of local BBS numbers.

Downloading is easy with *Interlink* automatically picking up on the name requested from the service and using it as a default. Beginners believe me, if I could learn to download source text and programs anyone can. *Interlink* is easy to learn the basics, but don't think this keeps it from performing many complex functions.

Automatic everything is available to the user. If you have never seen *Interlink*, it is worth getting a close look at. The auto-dial dialog box is unbelievable. 20 buttons on either side of a fully featured function box keep the name of BBS and friends numbers clearly displayed. Click on one, and the number is ringing. Click on auto re-dial and it will continually dial the number requested. Click group and the auto re-dial will try each member of the group. Click on autolog after selecting a name and it will let you customize several replies to question prompts. At the answer prompt the user may play a recording of keyboard strokes enabling powerful macros to be created for each BBS. The RS232 button pops up another powerful dialog box containing settings for baud rate, parity, wordsize, duplex, XON/XOFF, and stop bits. Everything is set with buttons, the use of 300, 600, 1200, 2400, 4800, 9600 or 19200 baud rates is supported. None, even, odd buttons let you select parity. Wordsize choices include 5, 6, 7, and 8. Select the style button from the telephone dialer dialog box and still more settings are available. Emulation modes of VT52 or translation can be selected. The translation table dialog box makes transferring from one ASCII to another simple. This feature lets you look at files from disk without taking action on special control codes. The (?????) button lets the user take advantage of new emulation modes developed without paying for upgrades. Some emulation types available include VT100 with PC ANSI supported, ATASCII for 8 bit Atari boards, and quickb for compuserve (look for more in the future). Settings include echo and linefeeds on input, output. special effects include 40 line screen and reverse status line. Features include toggle colors, wordwrap, and connect chimes. Selecting modem from the telephone dialer dialog box allows the user to set modem configurations like the initialise string, dial string, prefix/suffix, hangup string, and failure string. Also setting time out limit and auto retry limit is available here.

Disk commands (F3) is a dialog box with many disk utilities such as print file, show file, directory, rename, copy, create folder, delete, drive path, and format. **Directory** is nice to see how much room is left on disk. **Drive path** is especially nice, the specified path name can be set and then save the phone file, so next time you load the program your default is set automatically. The **format button** presents another dialog box with options for formatting the disk such as drive A or B, 1 side or two, format density 400K or 360K. All these extras make it a snap to use **Interlink**.

Transfer (F2) is what modem programs are all about. Once pressed or clicked on from the menu bar a dialog box offering all the necessities for file transfer appears. A display of status with errors, mode, and block number are continuously updated while uploading or downloading. Selection of Xmodem, Ymodem, or ASCII are available, but again our friend the (????) button which means new protocols will be available for free. Some new protocols include **multixy** and **batch y** modem. With **multixy** the user is free to start a download and leave **interlink** to run other software. **Send\Receive** buttons let you choose your function. Again there is a drive path button which will let you set default paths for sending and receiving files. The **F2** key is a very commonly used key by myself. My routine is to use the menu bar to get the auto-dial dialog box. Get online with the touch of a button and then play a macro (alternate or control F keys) or recording to get where the downloads are. Select a program and then hit the **F2** key. Click on receive and wait for the end of download chime sound.

Buffer size is user selectable. **Kandid Camera** has nothing on **Interlink**, every thing that happens on-line is automatically saved into a buffer for later retrieval. Is this a word processor or a modem program? The buffer pulldown from the menu bar includes open window, close window, save buffer, load buffer, clear buffer, print buffer, unfiltered, and show status. **Show status** is nice. It shows the size of the buffer and block, where the cursor is and total lines, and bytes used and bytes available. **Editor** has set block start, set block end, clear setting, delete block, move block, copy block, save block, print block, block to clipboard, clipboard to cursor, and clear clipboard (clip board features in a modem program) on it's menu drop down. **Finder** can find string, repeat find, set mark, goto mark, goto top, goto bottom, and goto block. I wish my word processor had all these features, I will have to ask Randy to put in a spell check and thesaurus. One point to note about the editor drag for setting a block is that it has a four way action. This lets you go up down left or right with complete accuracy. Keeping up with what has happened while on line is easy with all this power. The buffer lets you know when it has wrapped around (like many functions, sound is used to let you know).

Continuous status is provided both during normal activities and when you are on-line. The normal status display is visible during all activities even editing, and it has

baud rate, parity, wordsize, stop bits, duplex, linefeed, emulation, wordwrap, line status, buffer free, connect time, date, and time. I like the connect time and buffer free displays. When online the bottom of the screen has a one line status consisting of time online, time, idle, manual, baud, duplex, buffer free, and wordwrap. Keeping your self informed is certainly easy.

Have you ever had to type the same thing again and again? **Interlink** takes care of this problem. After the first time you hook up to a new service, look in the buffer. All the question and answer strings are there. Simply fill these into the autolog of that service. If this does not cover all you need to do an alternate method is to use a recording. The record drop down functions are: record on and off, off and undue last, play now or later, playback speed normal or fast, compare length of 2, 3, 4, 5, 6, 7, 8, 9, 10, or 15, wait time of 5, 10, 15, 20, 30, 60, 90, or forever seconds. Have you ever seen a section of a board that interested you but you have a hard time remembering how to get there? With recorder, just map out where you want to go in a BBS service once and then record it. The recording may be played directly from the log on answer string ex: ANSWER: ^C:\my_rec.rec. Features like this make tele-communicating a breeze.

Turn your computer into a BBS complete with three different access levels. Set auto from the extras drop down menu reveals a dialog box for setting up your BBS. High, medium, and low access string prompts are across the top. **Message directory** and **low access directory** are next. You can set a general greeting and then messages for high/medium/low access and bye. Set the ring string and the pickup string and time out. Calls answered and messages left are displayed along with a status of the answer mode box (as usual the regular status display is viewable). **Answer mode** comes in handy if you want to communicate with friends.

Another handy feature for you accountants out there is **call logging**. Once a log file name has been selected and call logging turned on, and the phone file saved, call logging is active. Call logging creates a file that can be accessed through a program supplied with **Interlink** to tell all about what time was spent and cost of sessions. **Call logging (F10)** will display many system defaults to set like capture buffer on, off, auto, buff wrap on, off, printer on, off, backspace on, off, completion tone on. **Off.call logging** is turned on and off by clicking a button, and the log file string is in the dialog box. **Interlink** keeps count so you don't have to.

I am sure that I have not touched on all of **Interlink's** buttons, or dragged out all it's gems. I only hope that if you have not seen **Interlink** that you will look before getting another terminal program. **Interlink** has been a pleasure to use, but a lot of work to review just describing everything it does has worn me out. **Interlink** truly brings power to the novice or the expert.

Reviewed by Joel Fradkin

Secrets Of Dungeon Mastery A Dungeon Master Hint Book

by Sir Edvarg the Unfailing
Translated by Tracy Raye Hickman
Software Heaven/FTL Games
P.O. Box 112489
San Diego, CA 92111
\$11.95 (76 pages)

Save for an excellent section naming and describing "Artifacts of legend and of power," this hint book is only moderately entertaining while attractively put together. It has an irritating disregard for the possessive apostrophe and a frustrating vagueness akin to the appendix on Magick in the original docs that make it very difficult for me to recommend. They also reverted to different spellings of certain words: shirikan, mana (manna onscreen and in docs), and magic (magick in docs). The majority of the book covers techniques that should have become ingrained naturally by the time any adventurer reached level three.

I would have liked lists on weapons, spells, and clothing detailing which are most effective in dealing with monsters and dungeon conditions rather than the sketchy remarks made in passing. What is the trajectory of a swing compared to a parry? Is there a special attribute to wearing Elven clothing as opposed to a fine robe? Is there a difference throwing a weapon with the cursor hand and throwing it from the weapons menu? (Of course, in some cases, this choice isn't possible from the weapons menu, but you CAN throw a sword ...) Is a loaf of bread more nutritious than a slice of a screamer? Does holding a wand or magick staff in the right hand really help in casting spells or can you leave it in your left hand or a pouch? What is the freeze life box really good for? None of these relatively basic questions is answered.

I was disappointed with this book. Much of it goes into detail about fighting techniques that are pretty self-evident after the first two levels. At the end of the book are four or five line hints about certain areas on certain levels and that's just about all there is to help you. I think my review of DM contained general hints for playing that this book spends pages expounding in a style that seems to be written for an "in" crowd if it was supposed to be funny. It does give you a feeling of a much larger, well-developed world from which we may expect DM2 to explore, and that is good, I suppose. But it doesn't quite go far enough if you are really in need of help.

Mapping is strongly suggested, and the reason given is that there is a vertical relationship between levels, amid their seemingly chaotic layout, that could help you navigate more efficiently. I still prefer the H&G (Hansel and Gretel) approach to leaving unnecessary items at critical junctures to mark a path for backward retreats or to mark an area that has been explored. It's

pretty difficult consulting a map when the purple worms are after you ...

Nothing is mentioned about paying attention to the details of this game. For instance, paying close attention to what occurs when you swing a sword can tell you the hit points you dealt out, similarly when your character is struck, hit points are shown above your character's icon, although both occurrences are swift and there's little time to let such things register, yet this could give you a better understanding of what is going on than reading this book. That is how you can learn whether a parry is more effective than a swing in a certain situation. Unfortunately, I don't believe there is a reading for hit points from magick spells.

There are better sources for hints and help than this book on the various networks, and of the ones I've seen, nothing is totally given away. For instance, there is a very good doc on early spells and a listing of the Hall of Champions heroes that allows you to compare their attributes and possessions before you start rather than wander back and forth among them for comparisons. I've read hints from users that still require some thought and the need to "be there" that are less cryptic than those in this book. If the book couldn't provide full maps of each level, partial maps could have been supplied, maps that were "damaged" during Edvarg's or someone else's quest.

As mentioned, I did appreciate the section on artifacts, some of which are available if you choose the right hero. After reading *Dungeon Mastery*, I started with a new party, which I call Gothmaog's Angels: Gothmog, who had the Cloak of Night (for what it's worth); Chani (resurrected to turn her mana level more toward priest level than her present wizard level, after all, Gothmog is a journeyman wizard), who has the moonstone mentioned in the Artifacts section; Wu Tse (also resurrected because I know using the shirikens -- proper spelling according to actual ninja I know personally -- will get her ninja levels back and her mana can go either way toward wizard or priest); and Sonja, whose mana is wretched but I wanted the best female warrior I could get with the party balanced more toward wizards and priests. I can recommend this party highly, although how the cloak and moonstone have helped is a mystery to me as yet. Here's another unanswered question: do you wear the Cloak of Night like a shirt or around the neck ala Sonja's choker or Chani's moonstone? Would Chani's moonstone be more effective if she held it in her hand?

Other heroes I can recommend is Wuuf, who has an empty flask so potion-brewing can begin earlier in the game; Zed, who is well-balanced in all skills; Wu Tse, who I personally believe is the ideal character combining high mana with fighting skills that can be developed quickly due to her possession of those throwing stars; Linflas, whose bow allows you to use arrows found in the early going and whose magick skills make up for

relatively small mana levels; and Tiggy Tamal, who has the highest mana level of any characters (and is perhaps also the weakest). I've had Syra and Elija in a party with Wuuf and Sonja, and it was a wretched time just keeping the lights on without using a torch, though I also feel these two are strong characters to keep in mind. Certain characters appear to be lost causes to my way of thinking and to the way in which this game depends most on mana-powered heroes: Leif the Valiant and Hawk the Fearless have good skills, but no weapons and low mana; Azizi Johari is well-equipped, but relatively weak; Stamm Bladecaster is a similar case as is Leyla Shadowseek and Alex Ander. Perhaps they would be challenging heroes to try after one has taken their best group to the finish, I don't know. I'm sure there is hot debate in this regard.

Well, I'm getting away from the real subject of this review and that is the book that doesn't really want to give anything away. To which I must reply: then what is the point of this book? For instance, in Rings of Zilfin, there are lists of weapons and spells that explain their powers and relative uses, ditto for Wizard's Crown and many other FRP games. These lists come with the docs and there are extenuating circumstances for their existence, of course. But a hint book should really be more than a cryptic extension of the docs. FTL, which has given us an absolutely spellbinding, highly addictive game, blew it with this book. If you really need help, look to the networks or your users group PD disks for help from other gamers who really know how to write hints and provide useful lists and maps.

Reviewed by Donavan Vicha

GFA BASIC Training Reboot Camp

MichTron
576 South Telegraph
Pontiac, MI 48053
(313) 334-5700
\$14.95

For the beginning GFA Basic programmer *Michtron* has just published a paper bound 260 page book entitled **GFA BASIC Training Reboot Camp**, suggested retail price \$14.95. A companion disk is also available for an additional \$12.95. This disk contains all of the books programs typed in plus a smattering of programming tools including ones that help with sound effects, loading **Degas** and **NeoChrome** pictures and setting the color pallet.

This book is written for the novice programmer and starts with the basics of GFA BASIC. The first section is devoted to explaining the GFA BASIC editor and keyboard layout. This is followed by nine chapters crammed with short, explanatory programs. Starting

with one or two line long programs leading into more advanced programs dealing with topics such as random numbers, data structure, simple animation, mouse control, and sound effects. At the end of each chapter is a short self test that checks your comprehension of the material presented so far. At the end of the book there are some valuable tables that show the screen positions for the **PRINT AT()** command for all resolutions.

The book is clearly written and follows a logical progression in presenting its material in a tutorial form. It is, however, a book designed for someone who has little programming experience. It does not cover many of the more complex GFA BASIC commands and cannot be recommended for the intermediate or advanced programmer, who would be better off buying the **GFA BASIC Book** (reviewed in the June 1988 issue of **ST Applications**). But if you want to get your feet wet in the wonderful world of programming with a fantastic language I can recommend this book as a valuable tool to help you learn the fundamentals.

Reviewed by Ron Schaffer

Subscription Problem?

We all make mistakes.

If you have a subscription problem, please call:
(707) 887-7879

or

write us at:

ST Applications
P.O. Box 980
Forestville, CA 95436



If possible, please have your mailing label available, as well as your canceled check if you are having problems with payment.

If moving, please give us both your old address and new address.

Thank You!

Moving?

Santa Paravia and Fiumaccio

TDC Distributions
Starsoft Development
(305) 423-1987
\$19.95

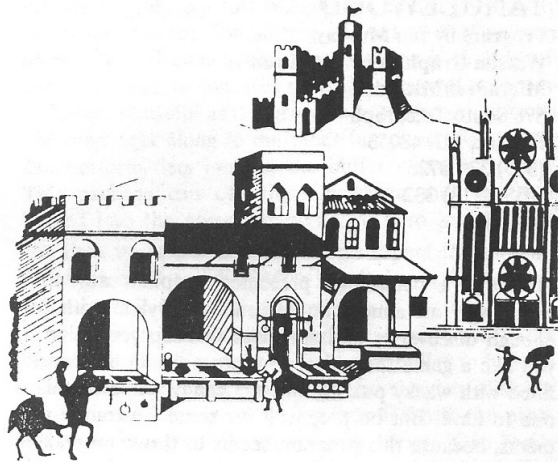
There are not many 'adult' computer games, and I do not mean those with sex and violence. I mean a game that has an adult theme and may teach you while it entertains you. **Santa Paravia** is such a game, a teaching game that is disguised as an economic situation. In very technical terms, you have eight interrelated partial differential equations with unknown constants that you must solve in your head.

The story of the game is quite easy. You are now the ruler of an Italian citystate in the year 1400. You are very ambitious and you want to change the state from a small agriculture community to an imposing mercantile empire. You must make wise decisions, treat your people with respect, invest in public works as well as take into account the mercurial nature of both the weather and the harvest. As part of the task, you can invite up to five of your friends to be rulers of neighboring states and make a comparison of your governing and administrative skills.

Each turn is played out upon a period of a year in the life of the city. There are four phases to each turn, the harvest, the tax, the public works and the display. In the harvest phase, you must take into account what you are planning to do with your kingdom. You must decide how much land to purchase, how much food to either buy or sell to feed your subjects as well as how much debt you feel that you can live with. As your debt increases, the possibility increases that your creditors will combine and strip you of your wealth as well as the large interest payments will become a very heavy load. You must also leave enough reserve grain for the next year.

In the tax section, you must decide how your kingdom is to be financed. You can levy sales taxes, income taxes and importation taxes. As with all taxes, you run into the Laffer curve. At some point when you increase taxes, the amount that you receive will decrease since it does not encourage people to perform the tax generating occupations. Of course if the taxes are too low, then you cannot support your kingdom.

With your tax money, You have many options on how to spend your income wisely or foolishly. You can spend the money to make new markets, which will attract merchants which generate income and sales tax. You can also build wool mills to help employ the serfs, but then they will not be available to harvest and grow the grain, so more money needs to be spent on food. You can also glorify yourself through the purchase of palaces and cathedrals, which are needed to increase your title from Sir to Baron, Count, etc, up to King. You can also increase your defenses, so that your less wealthy neighbors who might become envious of your

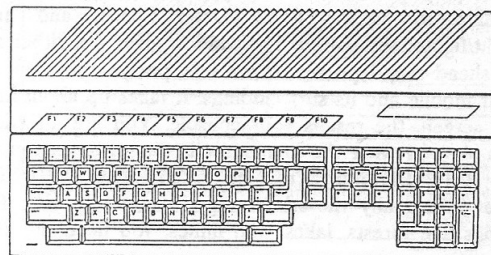


riches, will not see an easy road to wealth by conquering you.

Finally, you can see your kingdom and judge how well you are doing as compared to the other players. The whole game needs a combination of strategies. What works at the beginning of the game does not continue to work as the game changes and as your kingdom changes. Grain prices are determined by the law of supply and demand, as is the price of land. You can use your money to speculate in land, and hope that you can predict future prices in both grain and land.

The difficult thing to explain about this game is why it is so enjoyable. The various choices that need to be made, in different portions of the game, are made through colorful graphics screens that have dialog boxes and sections that can be clicked upon with the mouse. The different pictures are colorful, and unobtrusive and effectively move the players through the game. All of the pictures are loaded at the beginning of the game and to shift through them is very quick. The play is effective. There is even a screen that effectively summarizes how well you are doing on your quest to become king. Another good point is that this is a competitive game, with the ability to have up to six players, each of which will be called by name and the turns of the various players will be independent. The manual is poor, but overall this is a fun, effective game that will teach economics in a practical manner.

Reviewed by Sol Guber



TANGLEWOOD

Program by Ian Murray
Watson Graphics by Pete Lyon
Microdeal/Michtron
576 South Telegraph
Pontiac, MI 48053
(313) 334-8726
BBS: (313) 332-5452
\$39.95

Here is a beautifully presented graphics adventure game with an almost impenetrable storyline with just enough discoveries and surprises to keep you going. If you like a game that takes you into a whole new world, filled with wacky puzzles and eye candy graphics, this is one to have. But be prepared for some frustrating moments, because this program seems to thrive on making you pull hairs from your head.

You've been chosen by eccentric Uncle Arthur to save him from being claim jumped by a large, odious company that learned (from Artie, who was under the table at the time) about the rare Dog Crystals that can be found on the planet T'nglywd. These crystals have certain properties that could have military uses, so the company is setting the local galactic courts on Arthur, despite his rightful claim. He has ten days 'til his eviction and you are the only person he can trust to find the proper documents. Eventually you have to break into the company's highsecurity building, but first you have to learn how to operate the five mobiles which carry out your search.

The docs provide descriptions of the locals and hints about their rituals, which you must learn if you're to get their cooperation in your search. The docs are sketchy, which is in keeping with the whole scenario.

As the ninjas say, "keep on going," so must you if you're going to figure out the multiple layers of *Tanglewood*. Trust me on this, no matter which mobile you choose (one seems to be permanently out to lunch, so you really only start with four of them), it won't be the right one in the right place at the right time. Only one of them is immune to the disrupters that the evil company has sent out to hamper your activities and that one can only operate away from water at night. But they are such cute little buggies, and the world in which they traverse is so pretty that it is possible to get hooked into going on. This is a FULLY MOUSE CONTROLLED game.

The playing screen is fairly typical in structure, with a kinship to *Terror Pods*, *Deep Space*, and any flight/fighter simulation. The window gives you either an overhead map-type view or a side perspective view of your mobile and its surroundings. It takes up about half the screen, the rest is your console. You are afforded this view by what the manual calls the early version of the Ritter Pavlovsky Transfer so "what you see is not what is actually there." You get two viewpoints: for things like forests, lakes, and mines, you need this RP



transfer nonsense to see what's not actually happening, but what is a reasonable facsimile of what's happening. (You really begin to think this all makes sense after an hour or two of playing ...) The map view is a gorgeously drawn maze across the planet that is more than equal to the graphics of the first two games I just mentioned and much smoother in its scrolling.

Using the mouse to guide your mobile along crooked paths, over bridges, and through various mazes and fields in your search for various items that will help you breach the security center of the company, you get a Pac Man-like feel for the awkwardness of controlling the mobiles. Once you get used to it, the movement aspects of *Tanglewood* are secondary to its main cause of frustration.

You see, each mobile has certain deficiencies in its programming but they can be fixed if you find the right parts or programs scattered around this crazy world. You have to work very hard to be in the right place with the right mobile at the right time.

The frame of the mapping area is your console for controlling the five mobiles and their various functions, as well as game controls for saving games (up to 5 files per extra disk), starting and restoring, pausing, quitting, and turning sound on/off. Each mobile has a group of programs or data files that are accessed by yet another switch. Two more switches allow you to get brief descriptions of your surroundings or what your mobile may have discovered, and a time warp control that moves you from night to day and so on. (This is mostly for getting your best mobile off the lake and into action repairing your other mobiles.) There are heat gauges that go along with time of day, elapsed time (you have ten days ...), and a found object screen and three carrier screens.

This is a big, complex adventure, refreshingly original in its puzzles and graphics. Be prepared for frustration as well as fascination with this unique game from *Microdeal*.

Reviewed by Donovan Vicha

Switch/Back

by Alpha Systems

1012 Skyland Drive

Macedonia, OH 44056

(216) 467-5665

\$69.95

Switch/Back is a software and hardware package designed to let the user switch between two different programs which have been loaded into the computer at the same time. In addition it allows capturing programs to file and reloading them. And if this weren't enough it can also be used to archive programs, protection and all, so they can be saved as files.

There is both hardware and software involved in producing the desired functions. The hardware is a device the size of a gender changer that goes between the ST's printer port and the printer cable. The device has a button on top that must be pressed each time a function of the program is to be used. This could be a problem for someone who has the back of their computers covered by a stand or cabinet. The manufacturer claims that using the printer port does not effect normal printer functions. In use I did not run into any printer problems when using the computer as I normally do. Once, I did get a strange screen dump when trying to archive one particular protected file with **Switch/Back**.

The first function of this program I'd like to examine is that of switching between two programs that have been loaded into memory.

This is not "*multitasking*" and the manufacturer does not claim that it is. Two programs are NOT running at the same time. Only one program runs at a time. When you switch programs, whatever was going on in the old program you are leaving stops! I thought I'd emphasize the point so you are not disappointed by expecting something more.

One of the advantages of this program over products like *Juggler* and *K-Switch* is that it works with TOS as well as GEM files. It is also much easier to set up and run. Also, it seems to run more software than what I've seen of its competitors. It does not run all programs, and it is difficult to use on others. Those programs that load correctly seem to work well. In fact, I am writing this review with both *Print Master* (a TOS program) and *Word Writer* loaded in so I can switch between them. I can compose on both programs but of course I cannot print once I switch to the alternate program.

One of the disadvantages, of the packages use for switching between programs, lies in the fact that has nothing to do with the package. This has to do with running out of memory. I thought I'd never have that kind of concern again when I switched from an Atari 8 bit machine. I thought that again when I had my 520 upgraded to 1 meg. But here I am again needing more memory. It is specifically stated that this package requires 1 meg. of memory. But what one becomes painfully aware of is that most of my favorite programs with

accessories take up lots of memory and when I try to use them together, I'm singing the old "*Out of memory blues*". The documentation warns the user to strip off accessories etc. to conserve space, but many of my accessories are as useful as the program as the program itself.

Moving right along to the other functions this package can perform, lets look at the buffer capturing ability. This package can capture a program that has been loaded into the computer and save it to buffer which can then be saved as a file. Don't expect the ultimate pirating tool, because it just is not designed for that and it does not work well for that purpose.

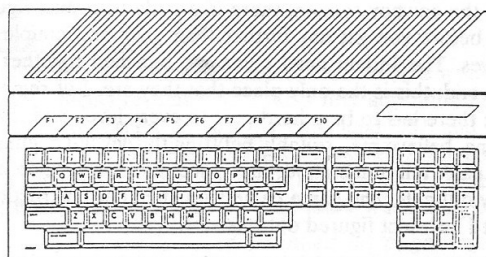
The memory inside the computer is partitioned off. About half of the memory is used as a buffer to hold an entire program. That buffer can be saved to disk for retrieval and with the programs included with the package it can be reloaded into memory or run. One handy use for this function is for saving your position. The documentation refers to this as the programs "*undo*" function.

Though you might think you can make file copies out of protected disks, most protection schemes are going to look to another part of the disk loading up. The documentation advises that you wait and save the program till after all the protection checking is done. That will work for some programs with simple protection schemes, but it does not for a number of programs I tried. Another problem that I should mention is that you have to load in the **Switch/Back** software first, so you cannot use many self-booting files. Most programs I'd like to consolidate seem to fall into this category, namely games.

One other function this program possess is the ability to archive programs as files. Again, if you're after a program that will circumvent protection schemes, this is not the program. To archive a program you must run a copy program like *Pro-copy* and capture computer memory before the write function is executed, and save that to disk. You then end up with a file that when executed writes that file, protection scheme and all, to a new disk.

I liked this program. It's virtues are a little bit exaggerated by marketing, but it is well thought out and easy to use. It does what it says it will do. My experience is that it does it less often than I'd like but it is superior to other products that claim similar functions.

Reviewed by Mikkell Lawrence



Rockford

Arcadia Software

Distributed by Electronic Arts

1820 Gateway Drive

San Mateo, CA. 94404

Ph: (415) 571-7171

\$29.95

Atari ST with Color Monitor and joystick???? If you are familiar with **Boulderdash**, an 8 bit game that also became an arcade hit, then you are already familiar with this game. Many parts of this game use the same ideas: the rocks falling on you, being able to stop the objects with your head, and trying to get to the end of a level. The things that make **Rockford** different from **Boulderdash** are the treasures you are trying to collect can also fall and kill you! Instead of just trying to get to the end of a level, you must solve many puzzles and collect the treasure along the way.

If you are not familiar with **Boulderdash**, then **Rockford** will be a real treat. It is a different kind of game, a type of **DigDug**, where you dig through the earth and try to collect objects and prevent rocks or other things from falling on you. Once you have collected the proper amount and your time has not run out; then you can go to the home base, and move on to the next level. However, on each level, there are traps as well as objects that try to touch you, causing you to die. Altogether, this is a puzzle game combined with an arcade game.

To make the game more interesting there are five different scenes. You have a choice whether to be a hunter, a cook, a cowboy, a spaceman, and a doctor. There are many differences between the choices. Each of the scenes can be selected when you start the game, and each have sixteen levels.

When you are a hunter, you try and get the pharaoh's death masks. You are in a jungle type motif with rocks that fall on you. The walls are made up of toucans, with the magic walls made of elephants.

In the cook's level, you try and find apples; with oranges being the objects that fall on you. The cook level is very deceptive as there are many things that trick you, like the bunches of apples all together, which are not as easy to get as they look. There is also a pizza monster which will chase you if you let it out of its cage. But there is also a good snake monster that will turn oranges into apples.

In the cowboy section, there is a code you must figure out before you get any further than your first couple of moves. The Indian head coins are bunched together at one end; this is the only place that they are. But remember: there has to be a hard part to every level, and there is one, believe me, suitably befitting this level: i.e. guns.

I can't tell much about the space man level, since it is very difficult. The game came with such bad instructions that I have not figured out this level.

The last one is the doctor level where you must be careful, very careful, because one extra block that you take out could mess it all up. But let me back up a little bit. In the doctor level, you must turn on two faucets, one that lets out water and one that lets out blood. When the two meet, they explode, opening the wall for you. But at the end, you face the grabbing hand.

Let me discuss the graphics and speed. The graphics are Ok, not the best that I have seen, but pretty good. Considering that, if they were any better, then they would not fit the game. Since this is a combination of a puzzle and an arcade game, the speed is a little fast. It is a pretty medium level game, with some of the puzzles being easy to figure out after four or five tries, but some are completely impossible, so it could take you years to figure some of them out.

Why do I like the game? Because, it is addicting! Once you play it, you know that you have to get better. Some levels you can pass and some you can't. That is just the way that it goes. The game is also attractive; when you look at it you just have to try out the levels even though some of them are complete impossible and you cannot figure them out. I really dislike the manual. It is four tiny pages of information about the levels, but the game is a good buy since it will make for many hours of play. I wish that when you have passed a level, you could get back to it without going through the easier levels, and also could save the high scores to the disk. I would recommend buying this game; if you can't see why, then read this review over again.

Reviewed by Rebecca Guber, Age 10

More Connect from page 62

of the files on your system. Conspicuously missing from the micro-DOS shell is the ability to run other programs.

Supported file transfer modes are XMODEM CRC and Checksum with optional adjustment for packet networks such as Telenet, and for slow host systems such as CompuServe. Simple ASCII upload or ASCII upload with a delay between characters is available.

You may run HI50.PRG to operate with a fifty-line display if you have a monochrome monitor. VT-52 terminal emulation is provided. The desk accessories cannot be accessed within **Connect**.

The documentation is excellent. Not only does it explain how to use **Connect**, it also has a good explanation of telecommunications basics and of the ST's file structure that will be appreciated by anyone wanting to learn more about computing.

Like **Chat**, **Connect** performs well. **Connect** adds a number of features to **Chat**, including editable function keys, the micro-DOS shell, and XMODEM CRC. If you like **Chat**, you definitely will want to get **Connect**. However, it is a very basic terminal program, and lacks features found in many public domain programs. Someone who is used to a more sophisticated program such as **FLASH** would not be happy with **Connect**.

Reviewed by Dennis Hevener

Questron][

Strategic Simulations, Inc. (SSI)
1046 N Rengstorff Ave.
Mountain View, CA. 94043
Ph: (415) 964-1353
Fax: (415) 961-6716
Telex: 989631 STRAT SIM INC
\$34.95

Questron][: Simplicity At Its Best

Questron][(Q2 from here on) is the latest fantasy roleplaying release from *Strategic Simulations, Inc.* I am certainly glad that SSI has chosen to release Q2 for the ST, considering the original game was never released for ST. While the game has made a step backwards in computer fantasy roleplaying game mechanics, it has made several leaps forward in the area of graphics. I predict that with the proper refinements, Questron][, should it ever be released, will be at least as good as *Phantasie*][.

Preparing for the Quest

The majority of the effort for Q2 went into the program itself, not the manual. While the box and manual were both "pretty," the manual is rather thin (14 pages) and left much too much information out. You find things out by playing the game, not in the manual. Another problem I had with the manual is that they used the back page as an advertisement for the AD&D products SSI will be releasing shortly. If you hadn't played the game, you might think that they simply released Q2 as nothing but an advertisement for *Pool of Radiance*. Once you see the superb graphics and user interface of Q2, you may become quite spoiled. While the graphics of Q2 are not quite up to *Dungeon Master* standards, I rate them an 8.5 on a 10.0 scale. In fact, the only problem I have with the graphics of Q2 is that the people and monster icons were not very well defined. While the landscape, buildings, cities, and 3D dungeons were done very well, I could barely tell that the icon representing my character was human.

The manual tells you that you should make a backup copy of the 2 program disks, before you start playing. I commend SSI for allowing owners of Q2 to make backup copies of the program, but must scold them for not providing for the hard disk owners out there. Q2 puts a dummy file on each of the two program disks AND on the character disk, with all three files having the same name. The reason this was done was obviously to make it easier for the programmer to tell which disk is currently in the drive (You heard me--you can't play from 2 drives either. Yuck!). If they would have put just a little more user consideration into their design, the programmers could have given the files .001, .002, and .003 extensions and, thus, allowed users to run off of either one doubled sided floppy, RAM disk, or hard disk. To make it worse, the four or so data files that Q2 requires you to create a character disk for, total less than 50K bytes.

The Game Itself

The philosophy behind Q2 is one that I embrace wholeheartedly: simplicity. It is not a huge game (a la *Ultima IV*), nor is it too frustrating (*The Bard's Tale*). It compares best with the original *Ultima*. One character setting out to complete one quest; improving himself along the way. Unlike most other computer fantasy roleplaying games, Q2 has your character predetermined. All of your statistics are set during your transport through time, which brings me to the plot.

Having stopped the evil magician Mantor and taken his book of magic in *Questron I*, you journey back in time to ensure that the evil book is never created. Your journey through time takes you to Landor, a circum-navigable world with two continents. You must find Mantor and his six cohorts and stop them from creating the aforementioned novel. There are essentially six types of adventuring areas: natural wilderness, towns, cathedrals, castles, tombs, and dungeons. Only the last three are very profitable, but you can have a lot of fun in the casinos in the towns of Landor. You will definitely want to try your hand at "Wizard's Squares..."

As I mentioned earlier, simplicity reigns in Q2. The selection of weapons, armor, spells, and commands reflects this simplicity. While there are few of each, it is important to remember that you do not need any more than what is given in order to complete the quest. For example, there are only four main spells: magic missile, fireball, sonic whine, and time sap. I have been getting along quite nicely with nothing more than fireball spells. In keeping with the simplicity theme, all commands can be accessed from the keyboard or via the mouse. This is highly commendable. In the game clues department, your first three miniquests should be the following:

- * Find the moonstone amulet and the bread of life.
- * Find the hall of visions and talk with your mentor.
- * Get to the Land of Sorcerers as soon as you feel ready.

The only complaint I have about the game play is that you must answer copyprotection throughout the game which becomes QUITE annoying. While this copy protection method is much preferable to disk protection, I wish that SSI would have implemented their protection more intelligently and simply. One or two questions at the beginning of the game would suffice. I hate it when I have to pull myself out of the adventure in order to look up the answer to some trivia question.

Conclusions

Once again, I emphasize that *Questron*][is a simple game. I highly recommend it for a first adventure game or as a refresher. I want to commend SSI and the authors of Q2, John and Charles Dougherty, for giving us a game with mild challenge. I find that life is challenging enough as it is. I don't need more frustration from

Continued on page ==> 73



New Software Etcetera

Should you have a product available for the ST, please send us a copy and we'll mention it in this column.

* indicates item currently available

Application & Design Software

226 NW "F" Street
Grants Pass, OR 97526
(503) 476-0071

* **Universal Item Selector II (Version 2.0)** (\$20) has features of Universal I plus Shift-Click and Rubberbanding Group functions, and more.

Electronic Arts

1820 Gateway Drive
San Mateo, CA 94404
(415) 571-7171 X590

* **A Question of Scruples** (\$49) this computerized version remains faithful to the original boardgame, while incorporating a database of over 230 questionable situations. Play with up to nine friends, or choose from sixty-four computer characters, each with its own personality and animated face that can be seen on the screen.

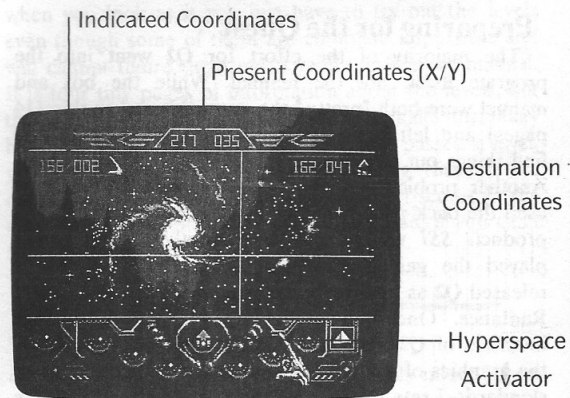
* **Warship** (\$60) a World War II tactical wargame set in the Pacific. Practically every warship that fought in the Pacific from 1941-45 is included in this game. Players can choose from 79 classes of warships from the Allied (Dutch, British, U.S.) or Japanese fleet, or modify ship data to create their own battleships. Players can also depart from the provided scenarios, and devise original maps.

Mindscape Inc.

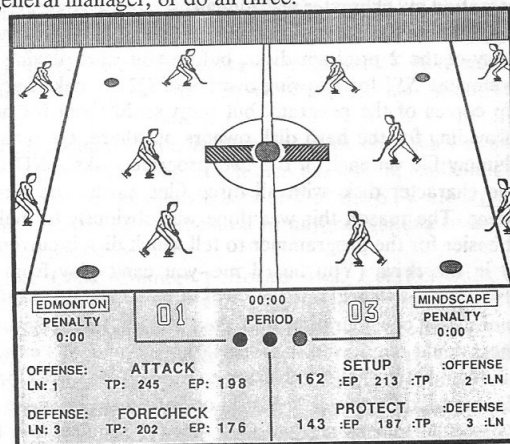
3444 Dundee Road
Northbrook, IL 60062
1-800-221-9884

* **Powerplayers Joystick** (\$30) high-tech engineered grip gives you greater control over any game you play, and makes every move you make the move you planned.

* **Captain Blood** the Hyperspace warp that sucked you into the game also fragmented your genetic structure, creating five exact duplicates of yourself. You must find your clones and assimilate them to replenish the vital fluids you so desperately need. The revolutionary graphics and visual effects, icon-based language for interplanetary communication, superb music and sound effects make this science-fiction adventure the computer game to end all computer games.



* **Superstar Ice Hockey** (\$50) speed skating, slap shots, hip checks, slick passes. It's all here. Play center or goalie, coach the team, build up the franchise as general manager, or do all three.



GAME SCREEN

New MIDI Software, etc.

MIDImouse Music

Box 877
Welches, OR 97067
(503) 622-4034

* **D-10/110 Capture!** (\$100) exceptionally powerful and comprehensive programming tool, making full use of the Atari ST's GEM interface.

* **D-50 Capture!** (\$100) editor/programmer/librarian.

* **MT-32 Capture!** (\$100) editor/programmer/librarian.

Dr. T's Music-Software

220 Boylston Street, Suite 306
Chestnut Hill, MA 02167
(617) 244-6954

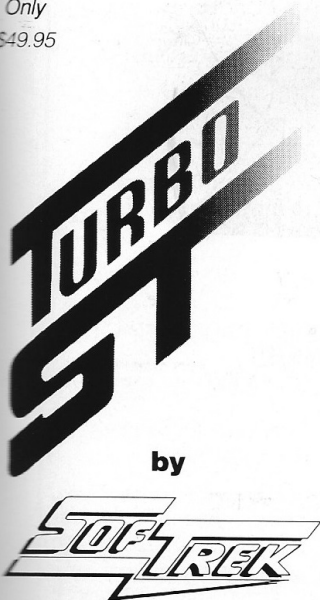
* **Keys Use MIDI** or not, playback your music through the Atari internal sound chip. Copy your favorite sheet music and hear what it really sounds like. Immediate musical notation displayed on the on-screen grand staff, and so much more.

More Questron from page 71

my computer games. I have played games such as **Ultima IV**, **The Bard's Tale**, and **Dungeon Master** and I enjoyed them for the most part. However, to many people (myself included), the constant mapping and solving of nearly impossible puzzles becomes tedious. **Questron II** allows you to get down to the adventure and bypass all of the bookkeeping hoopla. So, while I still consider **Phantasie II** to be my favorite simple adventure (and on my top 3 list for alltime favorite adventure), **Q2** comes very close to that #1 position for simple adventures. If you want something complicated that will keep you busy for months doing the bookkeeping (mapping, etc.) that the computer should do for you anyway, then **Questron II** may not be to your liking. Its emphasis is on simplicity and fun, not complexity and realism. For the rest of you, I highly recommend this game. Monochrome ST owners beware, however! This is a low res only game. When will software producers realize the benefits of producing games that work in all ST resolutions?

Reviewed by Anthony L. Farmer

Only
\$49.95



P.O. Box 5257
Winter Park, FL 32793
Phone: (407) 657-4611

The program you've all heard about ... Now better than ever!

- ★ Make your 520/1040 ST™ outrun a Mega ST™!
- ★ Speeds up the computer's response; saves time, reduces fatigue.
- ★ Low Cost; Less than half the cost of Atari's hardware blitter.
- ★ Not Copy Protected; Install on all your favorite program disks.
- ★ Works on any 520, 1040 or Mega ST™ with TOS™ in ROM.

Improved! Featuring Auto-install, more speed enhancements, and compatibility with more application programs than ever!

Turbo ST vs The Blitter (% speed increase)

	Monochrome		Color	
	Blitter	Turbo ST	Blitter	Turbo ST
dBMAN	10%	62%	8%	64%
Data Manager	83	81	85	80
1ST Word	37	29	34	34
GFA BASIC	22	69	13	62
Interlink	53	61	46	63
ST BASIC	221	400	219	425
Word Writer	34	29	35	32

To order, see your local dealer or send \$49.95 plus \$2 for shipping and handling to SofTrek. Visa/Mastercard accepted. Florida residents add 6% sales tax. Version 1.0 owners may upgrade for \$1 by sending in your original disk before Nov. 1.

Benchmarks performed by paging through an appropriate data file using Turbo ST™ on a 1040 ST and the blitter on a Mega ST. TOS, ST BASIC, 520 ST, 1040 ST, and Mega ST are trademarks or registered trademarks of Atari Corp.

Turbocharge Your ST Computer Today!

The reviews are in . . .

"A Best Buy' I'm impressed"

David H. Ahl, Atari Explorer, Nov-Dec 1987

"If you've got an Atari, you probably need this program."

Jerry Pournell, Byte Magazine, October 1987

"pc-ditto is a winner."

Charlie Young, ST World, July 1987

"This is the product we have been looking for."

Donna Wesolowski, ST Informer, August 1987

"This truly incredible software emulator really works."

Mike Gibbons, Current Notes, September 1987

NOW! RUN THESE IBM PROGRAMS ON YOUR ATARI ST.

Lotus 1-2-3
Enable
Sidekick
Crosstalk IV
EasyCAD
GW Basic

Flight Simulator
Ability
Superkey
Carbon Copy
DAC Easy Accounting
Managing Your Money

Framework
DESQview
Norton Utilities
Chart-Master
BPI Accounting
Silvia Porter's

Symphony
Q&A
dBase II,III,III+
Print Shop
Turbo Pascal
pfs:Professional
File

And Hundreds More!

pc-ditto is a software-only utility which expands the power of your Atari ST to imitate an IBM PC XT. No extra hardware is required (an optional 5.25-inch drive may be required for 5.25-inch disks). All your IBM disks will work "out-of-the-box".

pc-ditto features include:

- o both the 520ST and the 1040ST supported
- o up to 703K usable memory (1040ST)
- o not copy-protected -- installable on hard disk
- o imitates IBM monochrome and IBM color graphics adapters
- o access to hard disk, if hard disk used
- o optionally boots DOS from hard disk
- o parallel and serial ports fully supported
- o supports 3.5-inch 720K format and 360K single-sided formats
- o supports optional 5.25-inch 40-track drives

System requirements:

- o IBM PC-DOS or Compaq MS-DOS version 3.2 or above recommended
- o optional 5.25-inch drive is required to use 5.25-inch disks
- o 3.5-inch 720K DOS disks require a double-sided drive (Atari SF314 or equivalent)

*See pc-ditto today at an Atari dealer near you,
or write for free information!*

\$89.95

pc-ditto
by

Avant-Garde Systems
381 Pablo Point Drive
Jacksonville, Florida 32225
(904) 221-2904

Avant-Garde Systems, 381 Pablo Point Dr.
Jacksonville, Florida 32225 (904) 221-2904
Yes! Please send information on pc-ditto.

Name _____
Address _____
City _____ State _____ Zip _____